12/00/84
R. L. R.

(1)

RESEA-ACE
Reprint Articles from Other Newsletters

	Title / Author		Source / Issue
*	Alter DIR, Let's Talk Control		ACELO
	John David McFarland	.....	11/00/84 n
*	Bargain Corner		SBACE
	Daniel Prince	.....	11/00/84 n
*	Disk Directory, Creating a Different		ABACUS
	Unknown -	.....	10/00/84 n
	Display Lists		WAND
	John Keller	.....	11/00/84 n
*	Error 164, Elliminating		LAACE
	Ray Maynard	.....	12/00/84 n
	Fast Fire (Button)		SML-ACE
	Gerry Stotts	.....	11/00/84 n
	Fractals (Getting Down to Basics)		JACG
	Richard Kushner	.....	11/00/84 n
	Jumping between Playfield & Player/Misle		WLAAUG
	Gerry Wick	.....	11/00/84 n
	MX-80 Graftrax+ Atari Printer Codes		DAL-ACE
	Bob Dain	.....	11/00/84 n
	MicroFiler File Converter		SBACE
	Ronald Orr	.....	11/00/84 n
*	Musical Atari (A Piano Keyboard)		DAL-ACE
	Harald S. Poelchau	.....	11/00/84 n
	Musical Notes (part 5)		CURRENT notes
	Jay Gerber	.....	10/00/84 n
	Musical Notes (part 6)		CURRENT notes
	Jay Gerber	.....	11/00/84 n
	R-Verter Modem Adapter		SBACE
	Jeff Hogue	.....	11/00/84 n
*	Soft Touch		WAND
	Rolly Herman	.....	11/00/84 n
	Sound, Experimenting with		STARFLEET
	Lee Minard	.....	11/00/84 n
*	SynCalc		CURRENT notes
	Jack Holtzhauer	.....	11/00/84 n
	SynCalc: Serious Computing for Ataris		PACE
	Steve James	.....	08/00/84 n
	SynFile+		CURRENT notes
	Joe Waters	.....	11/00/84 n
	SynFile+		PACE
	Rick Gierl	.....	09/00/84 n
	SynTrend		CURRENT notes
	Allen Hart	.....	11/00/84 n
	TRAP Command, Atari Basic's		HAWKATARI
	Paul Maternowski	.....	11/00/84 n
	VisiCalc Quicky Guide		CHAOS
	M. Aldrich	.....	11/00/84 n

* OUT OF ORDER TO SAVE PAPER

DISPLAY LISTS

by Jean Keller

Several meetings ago I spoke briefly on the nature of display lists. I did not attempt to make a complete description of all of the hows and whys of display lists in the hope that some of you would try to use them and experiment on your own. On looking back on it I think that I was a bit hasty in my coverage of some of the aspects of the display list. Now that I am again actively programming, I would like to try to elaborate a bit on my earlier discussion.

First, remember that a display list is basically a set of instructions that tell the ANTIC chip how to set up the screen. In addition to telling ANTIC how to set up the screen, it tells ANTIC what memory locations should be displayed on the screen. It is possible to display any area of memory that you wish on the screen. You can even put your BASIC cartridge on the screen once you learn to fiddle with the display list.

Please refer to the GRAPHICS 0 display list that accompanies this article as I explain the structure of the display list (Fig.1).

To find your display list, type:

```
DL=PEEK(560)+256*PEEK(561)
```

Memory locations 560 and 561 are pointers to the display list. They tell ANTIC where to find the display list in memory.

The first three bytes of the display list (DL+0, DL+1, DL+2) are 112,112,112. Each "112" instruction causes 8 blank scan lines to be printed. These instructions cause a border to be printed at the top of your screen.

The next byte (DL+3) is the LMS (load memory scan) instruction byte. Although it appears to we humans as 66, to ANTIC it is really 64+2. The 64 tells ANTIC that the next two bytes form the first address of screen memory. The two tells ANTIC to begin displaying GRAPHICS 0 on the screen.

To calculate the actual starting address of screen memory, try the following:

```
SCR=PEEK(DL+4)+256*PEEK(DL+5)
```

Then clear the screen and type:

```
POKE SCR,255
```

A small inverse pointy character should appear in the first byte of screen memory, which is at the top left corner of your screen. These two display list bytes are the ones that you would change to have the computer display other areas of memory on the screen. The second number (DL+5) is the page number of screen memory. Adding 1 to this number will cause screen memory to begin 256 bytes higher in memory. Add 8 and you'll be displaying a portion of your BASIC cartridge on your screen!

After the address bytes come 23 more. Each 2 tells ANTIC to display one line of GRAPHICS zero. These 23 plus the one hidden in the LMS byte (64+2) result in 24 lines of GRAPHICS 0 being displayed.

The instruction 65 tells ANTIC to jump to the address formed by the next two bytes and wait for the vertical blank to end before beginning a new sweep of the screen. The address that ANTIC jumps to is the beginning of the display list. These two address bytes, therefore, are the same two numbers as the ones in memory locations 560 and 561 (since these memory locations "point" to the display list).

Since this list of instructions is in RAM memory, it can be modified. ANTIC will faithfully follow any changes that you make, so that you can mix graphics modes at will, provided that you adhere to a few simple rules.

A prime rule is that the display list cannot be too long, or data will be printed below your TV screen, out of sight. Even worse things can happen with very long display lists, so care must be taken not to exceed "the magic number". The magic number is, of course, 192 scan lines. Remember that a scan line is a narrow horizontal line across your TV screen. The entire screen is 192 scan lines high, so that each display list written by the Operating System in response to a GRAPHICS nn command has exactly 192 scan lines. (It is possible to be under 192; going OVER 192 presents problems.)

In a mode 0 screen, there are 24 horizontal rows of text. We will call each of these 24 rows a "mode line". Therefore, there are 24 mode lines in GRAPHICS 0. Since there are 192 scan lines, it follows that each mode line must contain 192/24, or 8 scan lines. You probably know that this is correct, since each character displayed on the screen is 8 bytes (scan lines) tall. It is therefore important to know how many scan lines are in each mode line when you are trying to build a display list. Careful planning will insure that you do not exceed 192 scan lines.

Another requirement is that you set up enough memory to hold all of the screen data that will be needed in your new mixed mode. The display list is located immediately before the screen data, which means that the screen data is sandwiched

CONTINUED ON NEXT PAGE

by

Polly Herman

Here are a few little ideas in BASIC that I find quite useful.

I like to add one or more screens of text instructions to various utility programs, etc. to remind me of what the program does and how to use it. In order to keep the text on the screen long enough to read it, some method must be employed to stop or delay the program. Here are some choices:

1. Use a delay loop such as;

```
10 FOR DELAY=1 to 500:NEXT DELAY
```

This is probably one of the first BASIC programming techniques that we all learn. It is alright, but it is difficult to get the length of the delay right for everyone reading and comprehension speed. Also sometimes the instructions suggest getting a pencil and paper and jotting down the pertinent keystokes. How can one judge the delay time to use?

2. A better approach is to use a method which allows the user to continue the program whenever he is ready. Here are a couple of ways to accomplish this:

a. At the place in your program that you want to stop, place an INPUT STATEMENT such as;

```
5 DIM HOLD$(1)
10 REM your text
20 REM goes here
30 REM and here
80 ?"press RETURN to continue";
90 INPUT HOLD$
100 REM program continues here
```

The DIM statement is necessary unless you are going to use a numeric variable and instruct the user to press only a number key. But then you would probably need an error TRAP.

In the above method, if you would like to get rid of the Question mark when the INPUT statement is executed, change line 90 to:

```
90 INPUT #16,HOLD$
```

b. An even simpler method is;

```
80 ?"Press any key to continue"
90 IF PEEK(753)<>3 THEN 90
```

Memory location 753 is known as KEYDEL. If it is equal to 0, no key was pressed. If it equals 3 then a key was pressed.

Try these, I think you will like them. ()

between the display list and the BASIC cartridge. You must start your display list far enough away from BASIC so that all of your screen data can fit. This is very easy, however. If you know which modes you would like to use, you can set the computer in the highest memory-consuming mode that you have before changing the display list. This will automatically move the display list lower in memory. (Remember that you will ALWAYS be able to find the display list using PEEK(560)+ 256* PEEK(561)).

Another thing to remember is that although our text mode is graphics mode zero, it is ANTIC mode 2. The display list therefore contains twos and not zeroes. You must always use the ANTIC mode number in a display list. The ANTIC mode numbers can be found on page 298 of the blue book "Your Atari Computer."

Try the following at your computer in immediate mode:

```
DL=PEEK(560) + 256* PEEK(561)
N=6:REM the ANTIC mode 0 for GR.1
POKE DL+10,N:POKE DL+11,N
POKE DL+12,N:POKE DL+13,N
```

Move the cursor into this area and type to see your modified display list. Try other values of N to see various other effects. Also experiment with SETCOLOR 2 and SETCOLOR 4 statements to see their effect on your display. Some very attractive title screens for programs can be built with a minimum of fuss by using this technique. I hope that I have been able to add a useful skill to your programming repetoire.

The listing of the display list is at the right. ------)

```
N=7:REM GRAPHICS 2
N=8:REM GRAPHICS 3
N=13:REM GRAPHICS 7
```

()

112
112
112
66
64
156
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
65
32
156

********** SPECIAL EVENTS

This month we extend congratulations to...

Leon Lazarus, Mario Taccariello, and Barry Courtade who celebrate their birthdays in December. Best wishes to all.

All events should be received by the 10th of the month in order for us to print it in the N.A.N.D. on time.

please send all items to:

N.A.N.D.
c/o Polly Herman
4 Charlotte Street
White Plains, N.Y.
10606 ()

****

(4)

FAST FIRE (BUTTON)

SAL-ACE NOV 84

by Gerry Stotts

Well, as promised last issue, here is your rapid fire article. You may build this into a joystick or use a male and female 9-pin "D" (joystick type) connector and use it with any joystick.

"I may be a computer dude, but remember: I don't want to be bent, folded or spindled"

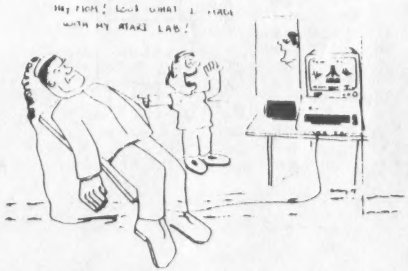
You will need an NE555 IC no matter which version you build (this may have a different number in front vis "NE"). This IC can be had from Radio Shack for $1.19 and elsewhere for about 50 cents. This is a timer IC. It puts out a continuous 'stream' of ones and zeroes if connected properly. This can be varied by the addition of a pot (or trimpot) instead of a 'fixed' resistor. Now if this is applied to the trigger-in pin of the computer, whatever program is running thinks you are pushing the fire button very rapidly! Now that we know what we are doing, let's continue.

"Go ahead and sue! It's your computer's word against our computer's word!"

Figure A shows the "pin-out" of the 555 IC. This is a top view (legs down, dot or notch on top). This is just so you know how the pins are numbered.

(Continued on next page)

"It must be you. The computer, it so happens, is user-friendly"



FAST FIRE

Continued...

Making a Standard "Fast Fire"

Figure B shows how to wire the IC with a male and female connector. If you are going to use this method, you will need a male 9-pin "D" connector (Radio Shack - $1.99) and a female 9-pin "D" connector (Radio Shack - $2.49). I suggest you use stiff wire and wire all pins of the male back to back with the female with the exception of pin 6. Wire the pins fairly close together (so female connector can go into computer and male is then facing out to accept a joystick). If you do this your fast fire will be quite short and won't cause the joystick connector to stick out from the computer very far (this is what is wrong with the commercial versions of this circuit). If you use this suggested method, you will be forced to wire your IC "dead bug" style. This means to mount the IC on top of your connectors (a little cement here will help) with the legs facing up. If you don't like working with the IC upside down, CCAREFULLY bend the legs out straight and glue it down right side up. Use "wire wrap" wire (28 or 30 gauge wire... very small) to do the pin 2 to pin 6 short. Use this fine wire to wire pin 3 of the IC to pin 6 of the female connector, and pin 1 of the IC to pin 6 of the male connector. Then fit the rest of the components around this and carefully solder them to the IC. Consult the parts list at the end of this article for values etc.

Another method of mounting would be to make an extension cable for your joystick with this mod somewhere in the middle. The disadvantage here is that your IC and your speed adjust pot (variable resistor) would probably end up on the floor most of the time... not very handy!

Making a Joystick "Fast Fire"

This method is a little cheaper but has a great disadvantage: you're stuck with using it in one joystick. Refer to Figure C. Notice that with the addition

(Continued in next column)

of a toggle switch, you can flip from "Fast Fire" to normal joystick operation. This switch should be a single pole double throw switch (SPDT). The center pin goes to pin 6 of the 9-pin connector. To build this version you may want to use a small piece of "perf board". This is circuit board material with a bunch of holes drilled in it (also available from Radio Shack). This may or may not be to your advantage due to the spare space inside your joystick.

(5)

As with the "Standard Fast Fire" you will want to use 28 or 30 gauge wire to hook this up (per Figure C). Use a multi-meter or other method to determine which wire on the fire button goes to pin 6 of the 9-pin connector. This is important so get it right!

All versions of "Fast Fire".

PARTS LIST	Radio Shack #
IC - NE555 timer	276-1723
C1 - 10 uF tantalum	276-1436*
R1 - 1000 ohms 1/4W	271-1321
R2 - 20K ohm pot or trimpot **	
S1 - SPDT switch	275-625***
- male 9-pin "D"	276-1537****
- female 9-pin "D"	276-1538****

* This is a solid tantalum 10 volt or so capacitor. Don't substitute an electrolytic!

** I used a Spectrol model 84-3-10-203. This trimpot will be very hard to find but is the nicest little pot I have seen for this particular purpose, especially if you build the 2 connector version.

*** This switch is only used on the joystick version

**** These connectors are used for the "Standard Fast Fire", not the joystick mounted version.

(Continued on next page)

Continued...

Please note that the IC is heat sensitive (like most IC's), so don't hold the soldering iron on it very long! Don't use a soldering gun!!! If this is all you have forget this project. Use an iron with a small tip.

Remember, wire all pins of the connector except pin 6 (standard version) so the joystick will function. Make sure the "+" symbol on the capacitor is towards the IC (or "-" symbol is away from the IC) as shown in Figure B and C. Other values of trimpot (variable resistor) can be used but C1 must then be changed to give a good range to the fire rate. If you use a larger value trimpot you can use a smaller capacitor. If you have a junkbox then experiment a little. If you want a slower rate you will either need to raise the resistance of R2 (pot) or capacitance of C1.

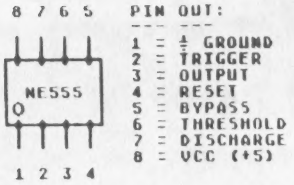
Finally, with the Standard Fast Fire, I used a small plastic box and cut slots in each end and slipped the connectors inside (prewired except for pot). This box was open on the bottom. With the Sprectrol pot, I drilled three small holes in this box. I pushed the pot into these and glued it then soldered it to proper pins. This made a very small package. It all measures about 1 1/4 inches (end of connector to end of connector). I made two of these, one with the pot on the left and one with the pot on the right for ports 1 and 2 on the computer!

Have fun! Don't attempt this unless you have some soldering or other hacker type experience. If you have problems, call or write. I'll try to answer them. Oh yes, I used Radio Shack part numbers only because they are usually available almost everywhere. I don't work for Radio Shack.

I'll bring these little jewels to the November meeting if I am able to make it. See ya there!!!

(Continued in next column)

FIGURE A



TOP VIEW

DOT ON IC SHOWS PIN 1

(SOME IC'S HAVE A NOTCH ON ONE END TO SHOW PIN 1)

FIGURE B

IC = NE555

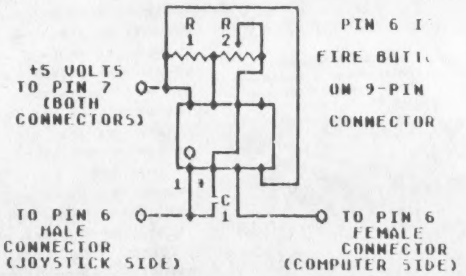
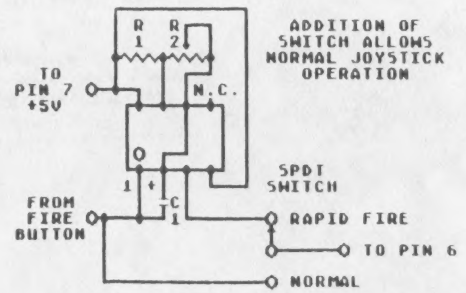


FIGURE C

FAST FIRE INSIDE JOYSTICK



FRACTALS JACG Nov 84

Getting Down to BASICs
by Richard Kushner - JACG

Anyone who has seen the Atari game "Rescue at Fractalus" has probably already been exposed to the world of "fractals". It is fractal mathematics that is used to generate the mountain ranges of this game. This is a big advance over the idea of plotting the mountains coordinate by coordinate and then keeping track of the movement of each point as the view shifts. From my reading, I have seen how fractals are being applied to mimicking many natural phenomena. They have the interesting property of retaining their general form as you look at more and more detailed pieces of them, that is, as you take a more and more magnified look at them.

My interest in fractals was sparked by an article and BASIC program in the September '84 issue of BYTE magazine. An Apple BASIC program was presented that permits one to see how fractal patterns are generated. I refer you to that article for an explanation of what the program does. Most of the work with fractals is done on very powerful (ie., Cray) computers and displayed on screens with at least 1000X1000 resolution (as opposed to the Atari's 320X196 resolution in GRAPHICS 8). But it is still fun to play around with these things and try out a variety of inputs to see what interesting patterns can be generated.

My adaptation of the program is given below. I've used GRAPHICS 8 to give the most possible space to plot points. The user inputs four numbers: LAMBDA (X,Y) which describes the fractal to be plotted, SCALE which determines how large the figure will appear on the screen and NUMBER which tells the computer how many points to plot. The text window at the bottom of the screen shows a running count of the number of points plotted. It is generally best to start with a SCALE value around 4, trying other values to plot interesting figures on a larger scale. Smaller SCALE values mean higher magnification. If the magnification is too large, many points will fall off the screen. The program takes care of values too large or too small to plot in lines 6002-6008, where these points are plotted on the screen edges. If your plots are flattened on their outer extremities, use a larger SCALE value. I also had to use the ABS function in lines 1020 and 1030 to prevent "negative squareroot" errors from crashing the program.

Also shown below are three respresentative plots and the LAMBDA (X,Y) values that were used to generate them. The first actually has a name, the dragon, and the second is a variation on that theme. The bottom one is a good one to show how the detail seen in the larger pieces propagates down to the smallest pieces you can see. I plotted 10,000 points to get these figures and this took quite a while (more than one hour each). One hundred points doesn't show up enough detail, but a couple of thousand points is really sufficient. Because the figures are pretty regular, with a degree of randomness, many points end up getting plotted and replotted many times. I encourage you to play around with this program and explore the world of fractals on the Atari computer.

Next month, we'll go back to explore the SINESCAPE program of last month. We'll show any interesting variations on the program that I happen to have sent to me and see if we can create some of our own.

```
10 REM FRACTAL PROGRAM BY GREG TURK
20 REM FROM BYTE/SEPT '84
30 REM MODIFIED FOR THE ATARI
40 REM BY RICHARD KUSHNER -JACG
50 CX=160:CY=80
60 X=0.5000001:Y=0
100 GOSUB 5000
110 GRAPHICS 8:POKE 752,1:COLOR 1
120 FOR I=1 TO 10:GOSUB 4000:NEXT I
140 GOSUB 6000
150 GOSUB 4000
155 COUNT=COUNT+1
156 IF COUNT=NUMBER THEN PRINT "ALL DO
NE!!!!":END
160 GOTO 140
999 END
1000 REM SQUARE ROOT OF X,Y
1005 T=Y
1010 S=SQR(X*X+Y*Y)
1020 Y=SQR(ABS((-X+S)/2))
1030 X=SQR(ABS((X+S)/2))
1040 IF T<0 THEN X=-X
1050 RETURN
2000 REM FOUR OVER L
2010 S=LX*LX+LY*LY
2020 LX=4*LX/S
2030 LY=-4*LY/S
2040 RETURN
3000 REM X,Y TIMES L
3010 TX=X:TY=Y
3020 X=TX*LX-TY*LY
3030 Y=TX*LY+TY*LX
3040 RETURN
4000 REM FUNCTION OF X,Y
4010 GOSUB 3000
4020 X=1-X
4030 GOSUB 1000
4040 IF RND(0)<0.5 THEN X=-X:Y=-Y
4050 X=1-X
4060 X=X/2:Y=Y/2
4070 RETURN
5000 REM SET VALUES
5010 GRAPHICS 0
5020 ? "WHAT IS LAMBDA? (X,Y)";:INPUT
LX,LY
5030 GOSUB 2000
5040 ? "WHAT IS SCALE ";:INPUT SC
5050 SC=2*CX/SC
5060 PRINT "NUMBER OF POINTS TO PLOT";
:INPUT NUMBER
5070 RETURN
6000 REM PLOT X,Y
6001 XX=SC*(X-0.5)+CX:YY=CY-SC*Y
6002 IF XX<0 THEN XX=0
6004 IF YY<0 THEN YY=0
6006 IF XX>319 THEN XX=319
6008 IF YY>159 THEN YY=159
6010 PLOT XX,YY
6020 RETURN
```

JUMPING BETWEEN PLAYFIELD AND PLAYER/MISSILE COORDINATES

by Gerry Wick

Recently I wrote a program in which I needed to know the exact correlation between coordinates in Graphics Mode 0 and Player/Missile Graphics. Every BASIC programmer knows that Mode 0 has 40 columns and 24 rows. It may not be so well known that Player/Missile Graphics has 256 columns and 128 rows in double-line resolution. (Some of them are off the screen and out of our range of vision.) But how do they overlap? To complicate matters, I understand that each TV or monitor has its own unique characteristics.

You may be wondering why you would ever want to know such information. Suppose that in a program you are writing you want to guide a dog around on the screen with your joystick. You created the dog using player/missile graphics and you want to enter one of several doghouses created on the screen using a redefined character set in one of the text modes. If the houses are identical except for their location, you need to know how to map the locations of the houses on the dog's coordinate system in order for the computer to know when the dog entered the correct house. A simple collision detection will not give you the answer if the houses are identical.

As another example, suppose I created a blinking cursor from a player and I want to place it at Mode 0 coordinates (20,20). Where do I POKE the player in its own coordinate system?

The program listing shows how to line up the two coordinate systems. It assumes that you know the basics of Player/Missile graphics. If not, there are numerous articles in Atari books and magazines that explain how to setup and use Player/Missile graphics.

The listing is for double-line resolution, single-width players. The horizontal (X) coordinate for the player is positioned at its bit 3, which is the center of the player in most applications. When RUN, the program will display a coordinate system in Mode 0. The player appears as a short bar (or dot) and can be moved with the joystick.

The coordinates of the player appear as text on the screen with "X position" for horizontal coordinate and "Y position" for vertical coordinate. Then the player can be lined up with the Mode 0 coordinates and comparisons can be made. Their relationship when plotted on a graph follows the equation:

X(P/M) = 4X(0) + 45

where X(P/M) is the horizontal coordinate in Player/Missile Graphics and X(0) is for Mode 0. For the Vertical direction the equation is:

Y(P/M) = 4X(0) + 17

To determine the equations for double-width players, change line 255 to POKE 53256,1. POKE in a 3 to find the equations for quadruple width. The graph shows that the horizontal relationships change, but that the vertical equation is the same for all widths.

In order to obtain the relationships for single-line resolution players, the following changes to the program are necessary:

230 POKE 559,62

270 change 512 to 1024 and change 640 to 1280

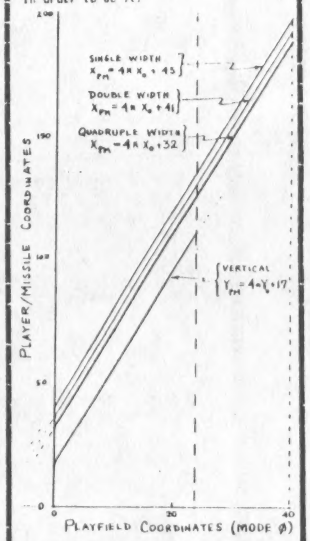
310 change 120 to 240

320 change 512 to 1024 and change 640 to 1280

If you run this modified program you will find that the relationship between the horizontal coordinates are the same as for double-line resolution. However, there is a change for the vertical equations.

Using these equations you can easily transfer back and forth between the two coordinate systems. To convert these equations to other graphic modes is easy. There is a straightforward ratio between the numbers of columns and rows in Mode 0 and all of the other text and graphics modes.

If you are ambitious you can figure out a way to modify this program so that the equations are automatically calculated and plotted on the screen, thereby avoiding plotting the equations by hand. Warning: you will need to know how to use the collision registers in order to do it.



LISTING

```
100 PRINT CHR$(125):REM CLEAR SCREEN
105 POKE 752,1:REM INHIBIT CURSOR
110 POSITION 0,11
120                                 PRINT
"01234567890123456789012345678901234567
89"
130 FOR J=0 TO 23
140 POSITION 21,J
150 PRINT J-INT(J/10)*10;:NEXT J
160 POSITION 2,2:PRINT "X POSITION"
170 POSITION 25,2:PRINT "Y POSITION"
200  J=PEEK(106)-16:REM  SETUP  POINTER
FOR PLAYERS
210 POKE 54279,J
220 PMBASE=256*J
230 POKE 559,46:REM ENABLE P/M
240 POKE 53277,3:REM TURN ON PLAYERS
250 POKE 704,60:REM PLAYER COLOR
260 X=120:Y=60
270 FOR I=PMBASE+512 TO PMBASE+640:POKE
I,0:NEXT I:REM CLEAR PLAYER
280 ST=STICK(0)
290
X=X+(ST=7)+(X<215):X=X-(ST=11)-(X>10)
300  POKE  53248,X:REM  MOVE  PLAYER
HORIZONTALLY
310
Y=Y+(ST=13)+(Y<120):Y=Y-(ST=14)-(Y>5)
315 RESTORE 330
320     FOR     I=PMBASE+512+Y-1     TO
PMBASE+512+Y+1:READ  A:POKE  I,A:NEXT
I:REM MOVE PLAYER VERTICALLY
330 DATA 0,8,0
340 POSITION 5,4:PRINT X;"  ":POSITION
28,4:PRINT Y;"  "
350 GOTO 280
```

EDITOR'S NOTE: Player-missile graphics are covered in the following sources:

Poole, McNiff & Cook Your Atari Computer Osborne/McGraw Hill pp. 307-324

Small, Small & Blank THE CREATIVE ATARI Creative Computing Press, 1983 pp. 39-45

Fox & Waite COMPUTER ANIMATION PRIMER Byte Books/McGraw Hill pp 239-272.

Stanton & Pinal ATARI GRAPHICS AND ARCHADE GAME DESIGN Arrays, Inc. 1984 pp. 111-201

Seyer ATARI PLAYER-MISSILE GRAPHICS IN BASIC Reston Publishing 1984 158 pp.s

ACEED NOV 84

LET'S TALK CONTROL

(Reprint The Escape -- Little Rock Atari Addicts)

About once a year, whether I need to or not, I enter DOS and rename a file to a filename already existing on the disk. If you've ever done this you know what the problem is. You've got two files on the same disk, both with the same filename. If you try to change one, DOS changes both. If you try to delete one, DOS deletes both. If you try to load once, you only get one but it's the first one in the directory.

Is there a way to recover the other file? The answer is yes. The program listed below is called ALTERDIR (alter directory). It will read the disk directory directly from the disk (without the help of DOS), allow you to change any filename, and write the altered directory back out to the disk. The machine language subroutine in the program is a disk handler allowing you to do all kinds of things to disk data. In this case it is set up to read the 8 sectors of the disk directory. After you have entered your changes the POKE on Line 2070 changes the ML routine to write the directory data back to the disk.

This is not a program you'll need every day, but the day you do you'll be glad it's around.

John David McFarland

ALTERDIR

```
1 REM ) THIS ROUTINE ALLOWS THE USER T
O
2 REM ) CHANGE THE FILENAME OF NORMAL
3 REM ) DISK FILES. IT READS AND WRITE
S
4 REM ) DIRECTLY FROM & TO THE DISK
5 REM ) DIRECTORY.
6 REM ) 04/13/84                    JDM
9 REM
10 KEY 50
20 G.100
40 PRINT:PRINT:PRINT"Potential fatal e
rror":PRINT"Try again":FOR X=1 TO 200:
NEXT X:G.2020
50 PRINT CHR$(125):PRINT:PRINT:PRINT"I
nished....":POKE 752,0:END
70 FL=1:PRINT CHR$(125)
71 PRINT"      "
72 PRINT:PRINT"entry #     filename
   size":PRINT
75 RET.
90 K=0:TRAP 92:INPUT K:TRAP 40000
92 RET.
94 INPUT #3,T$:RET.
99 REM
100 DIM DD1B$(1020),T$(12):DD1B$=" ":
DD1B$(1020)=" ":DD2B$=DD1B$
105 OPEN #3,4,0,"E:":POKE 752,1:POS.2,
7:PRINT"a moment please...."
110 FOR I=1536 TO 1536+59:READ BYTE:PO
KE I,BYTE:NEXT I:REM SET UP ML ROUTINE
120 PRINT:PRINT:PRINT"Insert disk in D
rive 1":PRINT"and press (RETURN)":GOS
.KEY
1000 REM
1001 REM ) DISPLAY ROUTINE (
1002 REM
1010 X=USR(1536,ADR(DD1B$)):REM READ D
ISK DIRECTORY
1020 J=0
1050 IF J>63 THEN 50
1055 GOS.70
1060 V=ASC(DD1B$(1+16*J,1+16*J))
1070 IF V=66 OR V=98 THEN PRINT J;"
....",DD1B$(6+16*J,13+16*J);".";DD1B$(
14+16*J,16+16*J);"    ";
1080 IF V=66 OR V=98 THEN PRINT ASC(
185(2*16*J,2*16*J))*256+ASC(DD1B$(1+
*J,3+16*J))
1085 J=J+1
1090 FL=FL+1:IF FL<17 THEN 1060
1110 POS.2,22:PRINT"Type entry # or
ETURN) ";:G05.KEY
1120 IF K=0 THEN FL=1:G.1050
2000 REM
2001 REM ) CHANGE AND WRITE
2002 REM
2020 TRAP 40:PRINT CHR$(125):PRINT"
 may now alter the filename
2030 PRINT"listed below.  Be sure t
eep the ":PRINT"proper spacing.":POK
752,0
2040 POS.2,7:PRINT DD1B$(6+16*(K-1),
+16*(K-1));":";DD1B$(14+16*(K-1),16
*(K-1));:PRINT CHR$(27);:PRINT CHR$(
)
2050 PRINT"---------  ---":POS.2,7:P
I CHR$(31);:PRINT CHR$(30);:G05.KEY
2055 IF ASC(T$)<65 OR ASC(T$)>90 THE
40
2060 T$(9)=T$(10):DD1B$(6+16*(K-1),16
*(K-1))=T$
2070 POKE 1543,00:X=USR(1536,ADR(DD1
)):G.50
5000 REM
5001 REM ) DATA
5002 REM
5070 D.104,169,1,141,1,3,169,82,141,
3,104,141,5,3,104,141,4,3,169,105,14
10,3,169
5080 D.1,141,11,3,169,0,141,255,6,32
3,220,24,173,4,3,105,128,141,4,3,144
5085 D.238,5
5090 D.3,238,10,3,206,255,6,208,231,
```

THE MUSICAL ATARI
(a piano keyboard)

by Harald S. Poelchau

The Atari can make noises of all kinds, and musical ones too. You can convert the Atari keyboard to a musical keyboard, imitating a piano (sort of). The following program shows a simple way to use the keyboard to play tunes.

The approach here is to use look-up tables to convert the keyboard character codes to the pitch codes used in the SOUND command. We first read the pitch numbers from C below middle C for 3 octaves and store them in array S. Next, one could fill a corresponding array with the character codes, and then test for the one that equals the key pressed. This procedure, however, takes too much time for the higher notes, and slows down the response. A better way is to use the key character code as the index for the position in array K which holds the index for the corresponding pitch code in array S. This assures equal access time for all keys and is much faster.

The keyboard is arranged such that the lowest row of keys represents the lower white keys of the piano. The third row continues with the higher notes. The keys in the row just above the lowest row contain the sharps and flats, or black notes on the piano. The same goes for the top row of keys. So, the Z, X, C keys would sound the low C, D, and E of the piano, and so on upward. S and D would be C# and D#. The keys with ',' and '/' are middle C and D, and the scale continues with the Q key. If you are serious about using this keyboard it might be useful to fashion a keyboard overlay to show the right notes.

Line 190 opens the keyboard for input. The main loop of the program from line 200 to 260 gets the key input and stops the sound of the previous note (200). 210 prevents an error 9, and 220 is all that is necessary to get the proper index of array S for the SOUND statement in line 240. At the same time we check for unassigned keys. For some added entertainment I am changing the screen color for each note, in line 230. That's all there is to it. Now try your Star Wars theme.

```
100 REM *** MUSICAL ATARI KEYBOARD ***
110 REM ...  by Harald S. Poelchau ...
120 REM ..... August 1984 .....
130 DIM K(90),S(37)
140 FOR I=1 TO 37:READ X:S(I)=X:NEXT I
150 FOR I=1 TO 90:K(I)=0:NEXT I
160 FOR I=1 TO 37:READ X:K(X)=I:NEXT I
170 GRAPHICS 18
180 POSITION 2,5:? #6;"MUSICAL KEYBOARD"
190 OPEN #1,4,0,"K:"
200 GET #1,A:SOUND 0,0,0,0
210 IF A>90 THEN 200
220 I=K(A):IF I=0 THEN 200
230 SETCOLOR 4,I,8
240 SOUND 0,S(I),10,10
250 GOTO 200
260 GOTO 200
270 DATA 243,230,217,204,193,182,173,162,153,144,136,128
280 DATA 121,114,108,102,96,91,85,81,76,72,68,64
290 DATA 60,57,53,50,47,45,42,40,37,35,33,31,29
300 DATA 90,83,88,68,67,86,71,66,72,78,74,77
310 DATA 44,76,46,59,47,81,50,87,51,69,52,82
320 DATA 84,54,89,55,85,73,57,79,48,80,60,45,61
```

MX-80 GRAFTRAX+ ATARI CODES

PRINTER FUNCTION	DEC CODE	I.P. CODE
BACKSPACE	CHR$(8)	CTRL-H
DOUBLE WIDTH (to end of line)	* 14	CTRL-N
COMPRESSED MODE	* 15	CTRL-O
Cancel COMPRESSED MODE	* 18	CTRL-R
Cancel DOUBLE WIDTH MODE	* 20	CTRL-T
Escape code and character	* 27	ESC-ESC
UNDERLINE MODE (N=0=OFF; N>0=ON)	<ESC>45N	<ESC>-N
Set line spacing to 1/8"	<ESC>48	<ESC>0
Set line spacing to 7/72"	<ESC>49	<ESC>1
Return spacing to default 1/6	<ESC>50	<ESC>2
Set spacing to N/216 (1<=N<=255)	<ESC>51N	<ESC>3N
Italic character set ON	<ESC>52	<ESC>4
Italic character set OFF	<ESC>53	<ESC>5
Reset special modes to default	<ESC>64	<ESC>@
Set LF spacing to N/72"(<ESC>"A"N)	<ESC>65	<ESC>A
Set form len to N lines (def.= 66)	<ESC>67	<ESC>C
Turn on EMPHASIZED MODE	<ESC>69	<ESC>E
Turn off EMPHASIZED MODE	<ESC>70	<ESC>F
Turn on DOUBLE STRIKE MODE	<ESC>71	<ESC>G
Turn off DOUBLE STRIKE MODE	<ESC>72	<ESC>H
Sets skip over perf to N lines	<ESC>78	<ESC>N
Resets skip over perf to 0 lines	<ESC>79	<ESC>O
Sets SUPERSCRIPT/SUBSCRIPT MODE	<ESC>83	<ESC>S
Format <ESC>"S"N, N=0=superscript, N>0=subscript		
Reset superscr.,subscr.,unidir.	<ESC>84	<ESC>T
Unidirectional printing (L to R)	<ESC>85	<ESC>U
DOUBLE WIDTH on till turned off	<ESC>87	<ESC>W

The above chart of EPSON printer codes was provided by Bob Bain. It is much more usable than the chart provided in the manual.

DAL-ACE COMPUTER FAIR ADMISSION: 1 TOY
(NEW TOY OF $3.00 VALUE OR +CASH)

MicroFiler File Converter
by Ronald Orr

The MicroFiler database program cartridge from Microbits Peripheral Products is an impressive file system program. It's widely advertised, including the Summer '84 issue of the Atari Connection magazine. It's the only filing system program I've seen that makes use of the cassette recorder, a practical proposition. However, it has one drawback: it uses a non-DOS file format for disks.

The following program converts data disks from MicroFiler (which uses one physical disk for each data file) to DOS files which can be used by Basic, Atariwriter or whatever. This particular program prints each field of each record on a separate line and skips a line between each record. If this does not suit, then "muck around" with lines 560 to 630.

The original program was written by Microbits Peripheral Products. I converted it to OSS Basic XL which cleaned up the structure considerably. The ability to recognize the fields within a data record was included. As to the use of Basic XL as opposed to plain Atari Basic, I believe that if you are at all serious about programming in Basic you will use Basic XL as its improvements (richer structure, functions and speed) are clearly superior to, but upwardly compatible with, Atari Basic.

MF2DOS

```
10 Rem MICROFILER TO DOS CONVERSION
20 Rem WRITTEN IN OSS BASIC XL
30 Rem PRECOMPILE : NO PARAM COUNT ON USR CALLS
40 Fast :Set 8,0
50 Rem GET 128 BYTE OPEN BUFFER
60 Dim Buffer$(128),Char$(1)
70 Dim File$(16),I$(10)
80 Buffer$=" ": Buffer$(128)=Buffer$: Buffer$(2)=Buffer$
90 Bufferadr=Adr(Buffer$)
100 Sector=5
110 Total=0:Index=128
120 Print "MicroFiler to DOS conversion"
130 Print "DOS disk in Drive 1"

140 Print "MicroFiler disk in Drive 2"
150 Input "File Name)",I$
160 File$="D1:"
170 File$=File$,I$
180 Rem READ TO END OF SCREEN DATA
190   While Total<880:Gosub 710
200   I=Asc(Char$)
210     If I=0 Or I=128:Gosub 710
220     Total=Total+Asc(Char$)
230     Else :Total=Total+1
240   Endif
250   Endwhile
260 Rem  DETERMINE RECORD LENGTH AND STEP PAST TABLE TO
RECORD DATA
270 Gosub 710:Nflds=Asc(Char$)
280 Print "Number of fields is ";Nflds
290 Dim Fldlen(Nflds)
300   For I=1 To Nflds
310     For J=1 To 4
320     Gosub 710
330     Next J
340   Fldlen(I)=Asc(Char$)
350   Print "Field ";I;" length is ";Fldlen(I)
360   Next I
370 Gosub 660:Reclen=I
380 Print "Length of record is ";Reclen
390 Gosub 660:Reccnt=I
400 Print "N of records is ";Reccnt
410 Dim Stg$(Reclen)
420   For I=1 To 3:Gosub 710:Next I
430   For I=1 To Nflds
440   Gosub 710:Gosub 710
450   Next I
460 Rem GET THE RECORDS
470 Open #1,8,0,File$
480   For I=1 To Reccnt
490     For J=1 To Nflds
500     Stg$=""
510       For K=1 To Fldlen(J)
520       Gosub 710
530       Stg$=Stg$,Char$
540       Next K
550     Rem STRIP TRAILING BLANKS
560           If       Right$(Stg$,1)="                    Then
Stg$=Left$(Stg$,Len(Stg$)-1):Goto 560
570     Print Stg$
580     Print #1;Stg$
590     Next J
600   Print
610   Print #1
620   Next I
630 Close #1
640 End
650 Rem # SUBROUTINE TO GET BCD
660 Gosub 710:I$=Char$
670 Gosub 710:I$=I$,Char$
680 I=Val(Hex$(Dpeek(Adr(I$))))
690 Return
```

continued on page 9 column 2

Page 5

Musical Notes

By Jay Gerber

Hello, fellow music enthusiasts, and welcome back to Musical Notes! This month we are going to finish up a few more points which will help you to transfer sheet music directly to your computer. We will also discuss how to make a limited Basic program that plays sheet music. And as always, We shall have a great time!

Last month I was explaining the concept of chords in music. As I stated last month, you can think of a chord as a series of notes, one atop another, that are played at the same time. These notes must have the same duration. Instead of going into some more theory about chords, it is safe to say that you will see alot of chords in all types of sheet music, so you should learn to recognize and use them.

One interesting aspect about chords (especially to the Atari owner with only 4 voices) is that the top note of the chord always (unless seperated from the melody line) contains the melody of the piece. For example, look at figure 1. This is a piece of sheet music that you might find in any music store. Notice that all of the notes are arranged in chords. If you were to play this on a piano, or similar instrument, you would hear the Cantina theme from the movie Starwars. If you were to play only the top note of each chord, you would still hear the Cantina theme. The lower two notes of each chord are just accompaniment, or harmony notes.

Figure 1

This is useful for typing sheet music that has more than four voices into the computer. If you just wanted to put down the melody line, and arrange or forget the chords, look for the single notes in (usually) the Treble staff, and take the tops off of all chords in that staff.

Well, I think you all should applaud yourselves! You should, finally, be able to read any piece of sheet music! Now, how to make the Atari computer play it.

There are several ways in which you can make the Atari play sheet music. One is to buy one of the several commercially available music packakages on the market, such as AMS II, and Music Construction Set. (These and others will be reviewed, along with tips and sample sessions, in later columns). Another way is to program a music player. In 6502 Asseably, or machine language, this can get very hectic, or tedious at best. You can try Logo, except it only plays in two voices, and is hard to work with musically. I never figured out Forth, Pascal, or those scientific languages, so the only logical choice would be the one that most of us are familiar with: Basic.

Before I discuss the Basic music program, let me say that although the easiest to use, Basic is also verrry slow! It is impossible, without the help of machine language routines, to have all four voices playing different rhythms at the same time in Basic. In fact, getting two voices to play different rhythms is tricky enough! For simplicity's sake, I have two programs that will let you play either a melody line, or a melody line with chords, provided that all the notes have the same durations.

Figure 2 is a one-voice Basic player. Let's figure out how it works by taking it apart line by line.

```
1 REM ******************************
2 REM * ONE-VOICE BASIC MUSIC PLYR *
3 REM *        BY JAY GERBER       *
4 REM *  MUSICAL NOTES, OCT. 1984  *
5 REM ******************************
10 DIM M(20,2)
19 REM ***READ IN MUSIC DATA***
20 RESTORE 10000
30 FOR I=1 TO 20:READ
X,Y:M(I,1)=X:M(I,2)=Y:NEXT I
40 REM ***PUT BASIC PROGRAM HERE***
50 GOSUB 1000
60 END
999 REM *MUSIC PLAYING SUBROUTINE*
1000 FOR I=1 TO 20
1010 SOUND 0,M(I,1),10,10
1020 DUR=200/M(I,2)
1030 FOR DELAY=1 TO DUR:NEXT DELAY
1040 SOUND 0,0,0,0
1050 NEXT I:RETURN
9999 REM ***MUSIC DATA FOLLOWS***
10000 DATA 72,4,53,4,72,4,53,4,72,8
10001 DATA 53,4,72,8,0,8,76,8,72,4
10002 DATA 72,8,76,8,72,8,81,8,0,8
10003 DATA 85,8,81,8,85,8,91,3,108,2
```


Line 10 is a dimension statement. The first dimension in the two-dimensional array named M is 20. This is the number of notes that is in the piece. The second dimension, 2, is the number of values for each note to be played. This must remain constant because every note has two playing factors: frequency and duration.

Line 20 tells the subroutine which reads in the music data (frequencies and durations of each note) to start at line 10000. Line 30 reads in two values, with X being the frequency for note I, and Y being the duration for that same note I. Then it stores it into our array called M, so it can be recalled instantly at any time.

Starting at Line 40, you can put in your own Basic program, and make a subroutine call (GOSUB 1000) anytime you want the music played. Line 1000 starts the playing routine. For pieces longer than 20 notes, you should change the second variable to whatever number of notes you are playing. (This is also the first dimension in the DIM statement in line 10). Line 1010 plays the frequency of the note that the FOR/NEXT statement is currently on. Line 1020 figures out a close approximation of the musical duration of the curent note. (If the duration number in the DATA statment is 1 then note = whole; 2 = half, 4 = quarter ...).

Line 1030 is what is called in Basic a wait loop. It will do absoultely nothing for the number specified by DUR, or the approximated musical duration of the current note. Line 1040 turns off the sound. Otherwise, notes with the same frequency would slur together like tied notes. Remember that once you execute a SOUND statement, the sound is on until you execute you turn it off with this statement, or hit SYSTEM RESET. Line 1050 tells the computer to go on to the next note, and when it hits the 20th, or last note, it will RETURN back to the point after GOSUB was called.

Line 10000 starts the music data. The first number in each pair is the frequency value; the second, the duration. Obviously the order of the pairs has to correspond directly to the piece of music. The statements are arranged in no particular fashion, except that all DATA staements fit on one 38-column line.

Figure 3 plays the same song with three note chords instead of single notes. It works the exact same way that figure 2 does, except it has to handle three frequencies to each duration. One thing you might notice is that it is slighly slower than figure 1. This is because that it takes time to execute each SOUN statement seperately.

```
1 REM ******************************
2 REM *THREE-VOICE BASIC MUSIC PLYR*
3 REM *        BY JAY GERBER       *
4 REM *  MUSICAL NOTES, OCT. 1984  *
5 REM ******************************
10 DIM M(20,4)
19 REM ***READ IN MUSIC DATA***
20 RESTORE 10000
30 FOR I=1 TO 20:READ
V1,V2,V3,D:M(I,1)=V1:M(I,2)=V2:M(I,3)=V
3:M(I,4)=D:NEXT I
40 REM ***PUT BASIC PROGRAM HERE***
50 GOSUB 1000
60 END
999 REM *MUSIC PLAYING SUBROUTINE*
1000 FOR I=1 TO 20
1005 FOR J=1 TO 3
1010 SOUND J,M(I,J),10,5
1015 NEXT J
1020 DUR=200/M(I,4)
1030 FOR DELAY=1 TO DUR:NEXT DELAY
1035 FOR K=1 TO 3
1040 SOUND K,0,0,0
1045 NEXT K
1050 NEXT I:RETURN
9999 REM ***MUSIC DATA FOLLOWS***
10000 DATA 72,91,108,4,53,72,91,4
10001 DATA 72,91,108,4,53,72,91,4
10002 DATA 72,91,108,8,53,72,91,4
10003 DATA 72,91,108,8,0,0,0,8
10004 DATA 76,96,108,8,72,91,108,4
10005 DATA 72,91,108,8,76,96,114,8
10006 DATA 72,91,108,8,81,96,121,8
10007 DATA 0,0,0,8,85,102,128,8
10008 DATA 81,96,121,8,85,114,136,8
10009 DATA 91,121,144,3,108,144,182,
291,108,8,81,96,121,8
10007 DATA 0,0,0,8,85,102,128,8
10008 DATA 81,96,121,8,85,11
```

These two programs will help you add music into almost any Basic program. They were kept simple so yo could easily modify and incorporate them into you programs. One last thing about the second progra though: it is uneven at times. This is due, again t the time it takes Basic to operate all three voices a the same time. This can be corrected by changing th durations or modifying the approximation routine.

Well, that's just about it for this month. In the coming months I will review, with actual examples o

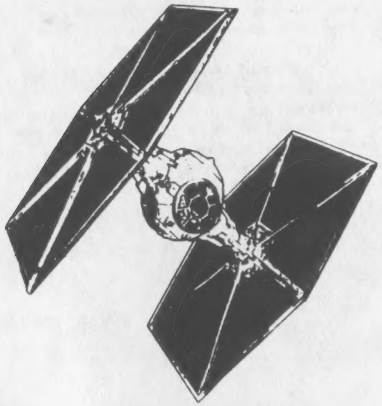
usage, the three major music packages for the Atari: Atari's Music Composer; API's ANS I and LotsaBytes' ANS II; and ELECTRONIC ARTS' Music Construction Set. I will also, after thouroughly explaining the music packages, get into some hard-core music theory, and show how you can improve your music to make it sound better, especially from sheet music.

To all you aspiring musicians: keep those music files coming. To those of you who have not sent or given to me (at meetings) your favorite music files -- why not? Send them to: Jay Gerber, 3639 N. 36th Road, Arlington, Va. 22207. You can also drop them off at either the Novatari or OC meetings.

Oh, before I forget .... Here are the answers to last month's chord-finding quiz:

G (G/D/B); G (B/D/G); C (C/E/E); G (G/D/B); G (D/B/G); C (C/G/E); G (G/D/B); G (D/B/G); F (C/A/F); D (A/F#/D)

It was pretty tricky since I put all of the chords in the Bass clef, and figure 3 (last month) was in the Treble!! Well, this month you can practice entering sheet music into the Basic program, and as always, HAVE FUN!



ABACUS OCT 84

Creating a Different Disk Directory

(The following is reprinted from ACCT (April, 1984), the newsletter of the ATARI Computer Club of Toledo. Author unknown.)

Thanks to Fred Choske for an answer to last month's question, "How can you create a disk directory for your programs other than in the standard location of sectors 361-368?" He referred us to a book from ALPHA SYSTEMS by George Morrison called ATARI Software Protection Techniques. Basically, the method is as follows:

1. Back up your completed disk.
2. Copy the directory to a new location.
3. Alter DOS to point to your new directory. DOS is stored in memory. Location 4226 normally contains 105, which tells DOS to look to sector 361 for the Directory. Compute the new value = 105 + (new directory sector number - 361). Then POKE 4226, new value.
4. Write the altered DOS files to your disk.
5. Destroy or change the old Directory, VTOC and DUP.SYS file.

The book, although small, contains a number of ways in which software is protected, and some small programs to enable you to accomplish this protection.

Musical Notes

Jay Gerber PART 6

Before we go on to a review of all the ATARI music packages, let's digress to a discussion of music and its history.

Music is the art of controlling and organizing sounds into intelligible patterns. The origin of music is unknown, and it is as mysterious as the origin of mathematics, art, or other abstract subjects. One might suppose that the first intelligible pattern of sounds may have been made by primitive man, banging wooden clubs to a rhythmic beat. Perhaps in accompaniment of this banging, the caveman started grunting on selected beats. Then another one started scraping the wall of the cave, and so this primitive form of noise would eventually become music as we know it today.

The first theories about music were developed by the ancient Greeks. They developed the present day musical scale of the related frequencies which I presented earlier. Pythagoras, the Greek mathematician whose triangle theorem is widely used, is believed to be the first person to devise a instrument to play the tones of the set musical scale.

Pythagoras found that the frequency of a note depends on the number of vibrations an object makes. By taking a metal wire and stretching it between two nails embedded in wood, he created what is known as a monochord. By plucking the wire, a tone is produced. Pythagoras learned that if he shortened the vibrating length of wire by pressing the wire down on the monochord, a higher pitch was created. By moving his finger up and down on the wire he shortened or elongated the wire, thus producing higher or lower tones.

Pythagoras soon discovered that certain tones are related to one another, and that when these notes were played in a certain progression, different moods could be created. For instance, a certain set of notes played in sequence could create a funeral dirge, another a happy, celebratory song.

The Greeks developed music into an art form, and devised many theories that are still used today. (We will be getting into the basics of these theories in later columns.) The Romans, being more interested in killing each other, did little with music. The next developement of music took place in the middle ages with the coming of the Crusades. Church music developed from a series of chants to hyms, played with, in the later part of the period, an organ.

It is during this period when music was first recorded on paper. Before, melodies and harmonies were memorized and handed down from generation to generation through the years. The memorization process was not an easy one. Choir members had to remember both the words and notes associated with them to all the psalms in the Bible. In order to make the task easier, Guido d'Arezzo, an 11th century Italian Choirmaster, developed a system to write and read the musical notes in order to save time. He developed what would eventually become the musical staff.

He sung the lowest and highest notes that were used by his choir. Then he wrote out a set of seven lines placed parallel to one another. He dipped the square end of his pen into ink, and marked the bottom line with a square. Then he sung the lowest note again. He then sung the next higest note he knew of. He placed this mark in the space between the 7th and 6th lines. He kept adding notes until he got to the mark on the top line, which was the highest note that was sung in the Psalms.

It was not until the late 17th century that the staff developed into the one used today. It was at this time that harmonic theory was developed. It was also the time of the great classical composers. All of the music we hear today, including rock, country, march, western, or whatever, contains techniques developed by Mozart, Bach, Beethoven, and the other composers in the late 17th and early 18th centuries.

Well I hoped you liked this dissertation. Next month, I will discuss the ATARI music packages just in time for Christmas. Hope to see you Bach next month!!!

* * * * *

SBACE 16 GAZETTE

Vol. 3, No.5

R-Verter Modem Adapter Review
by Jeff Hogue

This text has been transmitted to you by my Atari connected to a Hayes 300 baud modem by a R-Verter modem adaptor from Advanced Interface Devices (AID). This device, which lists for $50, looks like an familiar Atari serial bus connector attached to a wire going to a male DB25 (RS-232) connector. There are a few microcircuits in the DB25 but they get their power from the Atari.

There have been a number of general purpose devices which substitute for the printer function of the Atari 810 unit but this is the first one I know of which replaces the serial function. There are a few modems and printers which connect directly to the serial bus, and I'm using a typical disk drive (Trak- see my related review) with a printer output port. I wanted to be able to use a stand alone modem with my Atari and my Compaq (IBM PC clone) and to have a direct (non-modem) way to transfer files between the two without spending a ridiculous amount for an 850.

As you can see, it works and it certainly reduces the mess of wires associated with an Atari. On the other hand, it's taken me a little while to get here.....

The device works by replacing the Atari serial handler loaded during a correct power-up sequence from an 810 with an AID-provided handler loaded, say, as an AUTORUN.SYS file at boot-on. You need a MEM.SAV file also which really slows down getting in and out of DOS.

Actually, the worse problem I ran into was with the version of Amodem which I was using. It had some sort of weird timing problem and just didn't work. The AID people eventually equipped me with a special version which works correctly.

I don't know of any local place to buy the device, but I've seen it advertised in the Atari-oriented magazines. AID needs to know what class of modems you intend to use it with since the RS232 "Standard" seems to come in a lot of flavors.

AID has been good to perhaps overwhelming with technical help and they are even putting in a Bulletin Board for purchasers, if you can afford a call to Florida.

I think it is a good way to minimize the cost and physical size of an Atari system while keeping a good deal of flexibility. It hasn't been the most straight-forward computer problem I've worked with but I seem to have a solution in hand and I'll be glad to help you through the S.B.A.C.E. users group if you decide to give it a try.

BARGAIN CORNER
BY DANIEL PRINCE

Here are some bargains I have found this time:

George Beekman has a few of his very high quality 48k boards left for $75. Computer Mail Order,P.O. Box 66449, Stateline, NV 89449, 800-648-3311 has the Indus GT disk drive for $279 +3% shipping.

The following information is courtesy of Aaron Todd:

Many ATARI parts, including complete boards and service data are available from at least two sources in Calif. I have ordered from American TV Sales, 15338 Inverness St., San Leandro, CA 94579 415-352-3787. I found them pleasant and quick on service. Tell owner Ralph Maddox that you read this in the SBACE news letter.

Another possibility is Centurian Enterprises Box 3233 San Luis Obispo, CA 93403 805-544-6616. I have not ordered from them, but when I called for a catalog, it arrived in two days. !!AARON!!

continued from page 8

```
700 Rem ## SUBROUTINE TO GET CHARACTER
710 Index=Index+1
720   If Index=129:Index=1
730   Poke $0301,2:Poke $0302,$52
740   Dpoke $0304,Bufferadr
750   Dpoke $030A,Sector
760   Dummy=Usr($E453)

770   Sector=Sector+1
780   Endif
790 Char$=Buffer$(Index,Index)
800 Return
```

I work with computer systems at work but don't get hands-on experience there. I "mess around" with my two Ataris (800 & 1200XL) at home. I wrote the conversion program to feed the MicroFiler data into Atariwriter to be able to paginate the printout from MicroFiler.

Ronald Orr
168 Hill Street
Hermosa Beach CA 90254
(213) 379-2513

STARFLEET 11-84

EXPERIMENTING WITH SOUND
by Lee Minard

You've already heard me complain about the lack of realism in the sounds you get fom the instructional books....and the ATARI can do wonders with sound...so here's another of my attempts to get more out of my 800XL!

This time, I'm working on the sound of a train. Why you ask? Well, I did it for the challenge. And I learned a lot from it. Hope you do too.

This is an attempt to build the sound of a steam locomotive pulling out from a station. I got close. I was never satisfied with the whistle sound...I'd love to hear how you improved it.

```
10 REM TRAIN SOUND
20 REM BY LEE MINARD
30 REM STARFLEET, DENVER
40 GRAPHICS 2
45 POKE 752,1
50 B=10:H=1
60 REM STEAM RELEASE ***
70 SOUND 0,1,8,4
80 FOR DLAY=1 TO 50:NEXT DLAY
90 FOR STEAM=12 TO 6 STEP -0.5
100 SOUND 0,1,8,STEAM
110 FOR DLAY=1 TO 50:NEXT DLAY
120 NEXT STEAM
130 REM CHUGS ***
140 FOR G=15 TO 1 STEP -0.25
150 FOR A=6 TO 2 STEP -0.5
160 SOUND 0,B,8,A
170 SOUND 2,B+1,8,A
180 NEXT A
182 REM SPEED UP CHUGS ***
190 FOR DLAY=B*2 TO 150:NEXT DLAY
195 FOR DLAY=1 TO 25:NEXT DLAY
200 B=B+2
205 REM GRAPHICS FOR "TRAIN" ***
210 H=H+1
220 IF H=4 THEN H=1
230 IF H=2 THEN GOSUB 440
240 IF H=3 THEN GOSUB 460
250 IF H=1 THEN GOSUB 480
270 IF B=90 THEN 290
280 NEXT G
285 REM WHISTLE ***
290 FOR X=1 TO 100
300 IF X=12 OR X=14 OR X=65 OR X=69 THEN SOUND 1,75,10,5:SOUND 3,50,10,4:SOUND 2,52,49,4
310 IF X=13 OR X=16 OR X=68 OR X=70 THEN SOUND 1,75,10,1:SOUND 3,50,10,1:SOUND 2,52,10,1
330 IF X=19 OR X=73 THEN SOUND 1,0,0,0:SOUND 3,0,0,0:SOUND 2,0,0,0
340 FOR A=9 TO 1 STEP -0.4
350 SOUND 0,86,8,A
360 NEXT A
361 ? "CHOOO ";
370 NEXT X
380 FOR A=9 TO 1 STEP -0.4
382 SOUND 0,86,8,A
384 NEXT A
385 REM FOR DLAY=1 TO 2:NEXT DLAY
386 ? "CHOOO ";
390 GOTO 380
440 ? #6;"TRAIN";
450 RETURN
460 ? #6; "[illegible]";
470 RETURN
480 ? #6; "[illegible]";
490 RETURN
```

***

Syzygy

Syzygy is the name Nolan Bushnell found Atari under in the early 1970s. It was t first company he formed and it produced t now famous Pong game, the grandfather arcade games.

WHAT GOOD IS THIS TELEPHONE IF I CAN'T CONTACT A [illegible]!

SynCalc: Serious Computing for Atari Computers

By Steve James

By now, you have probably seen the new advertisements for Synapse's new applications programs for the Atari. A review of the powerful new database program, SynFile+ will appear in a future issue of Keeping PACE while this article will review SynCalc, an advanced spreadsheet program. The third applications program: SynTrend will be reviewed as soon as one of our members obtains a copy.

Syncalc is, without a doubt, the most advanced spreadsheet to become available for Atari computers. Like any spreadsheet, SynCalc lets you work with rows and columns of numbers. This makes spreadsheets useful for budgets, stock portfolios, scientific data, financial reports, loan schedules, and other applications that require handling a lot of numbers. Spreadsheets function as electronic calculators with large memories, so people have begun to use them for "What If?" analyses of a particular problem. For example, you could set up a spreadsheet to look just like a Form 1040 that you get from the IRS. However, the spreadsheet also allows you to program it by inserting the same formulas that you would use to figure your taxes, such as "subtract line 20 from line 19." This means you only need to enter the raw numbers into your form and the spreadsheet calculates all of the results instantaneously. This makes it easy to try several cases and record the results. People often call this exercise a "what if" type of analysis.

Historically, Atari owners have had a limited choice in spreadsheet programs. About the only programs on the market were the original Visicalc, at $150+ and HomeCalc, a very limited version of Visicalc for about $40. Most other computers, including home computers have a much wider selection of spreadsheets, some far more powerful and flexible than Visicalc. This always puzzled me since there is a well known saying that Visicalc sold more Apples than Apple sold Visicalc programs. Anyway, my longings for a truly powerful spreadsheet were satisfied when Atari began marketing Synapse's new SynCalc program. SynCalc lists for $99, which is less than Atari Visicalc, yet far outperforms Visicalc.

SynCalc offers you the choice of menus or commands. This is a big improvement over Visicalc which only supported command keystrokes, such as /IR to mean "insert a new row at the place in the spreadsheet marked by the cursor." The menus let you learn the commands at your own pace or permit you to never worry about memorizing command sequences. When you load SynCalc you see the message at the bottom of the screen: OPTION = MENU. Pressing the option key brings up a menu window at the bottom of the screen with the four main command categories: LOAD/SAVE, for loading and saving data to diskette; TEXT, for entering text headings of any kind into the spreadsheet, NUMERIC, for entering numbers and formulas into the spreadsheet; and COMMAND, for using any one of a multitude of operations such as formatting your spreadsheet, copying parts of the worksheet, clearing out errors, printing your results, etc. You simply use the cursor keys to move the cursor over one of these menu entries, type return and you immediately go into a sub-menu. In each sub-menu, you do the same thing, positioning the cursor over the item you want and typing return to execute the command. This moves very smoothly and when direct input is needed from the keyboard, the menus prompt you for it. Error trapping is excellent and SynCalc rarely lets you do something destructive without asking you if you are sure of what you are doing.

Like Visicalc, the computer displays a 40 column window into your spreadsheet, which has 255 rows by 128 columns. Moving the cursor will let you scroll smoothly vertically or horizontally though the worksheet. You can also go directly to a certain location by defining its "coordinates." SynCalc uses the popular format of letters for columns and numbers for rows: F25 means the location at the intersection of row 25 and column F. Locations in spreadsheets are usually called "cells" and the cells can contain text, numbers, or formulas. SynCalc lets you format either the entire spreadsheet or blocks of cells, or individual cells so to display your information exactly the way you want it. In the table accompanying this review, I have listed the format commands available with SynCalc. You can also set the width of each column individually; in Atari Visicalc all columns had the same width. All of the format options can be called through the easy to follow menus.

Syncalc supports a host of mathematical, statistical, and financial functions. A list of its functions appears in the accompanying table. Using these functions in your formulas unleashes the power of an electronic spreadsheet. For example, you can enter a column of data and have SynCalc automatically calculate the minimum, maximum, range, average, mean, standard deviation, and variance of the entries.

Continued on page 9

Continued from page 8

If you then find you have made an error, just type in a new value where the erroneous value was and voila...SynCalc immediately gives you the results.

All of the new Synapse programs were designed to share information with each other. You can pass data to and from SynFile+ and SynTrend. Visicalc worksheets can be converted to the SynCalc format, so you can easily upgrade to SynCalc and save all of your old worksheets. SynCalc passes data to and from the other Syn-series programs using the Data Interchange Format (DIF), so you can transfer data to and from other programs that use this format (B/Graph, Visicalc, etc). Furthermore, you can save your worksheet to disk as a text file for merging with a word processor, such as Atariwriter. With Atariwriter, you can insert a worksheet in a document by first saving the worksheet in text form, then loading Atariwriter, typing your document and positioning the cursor where you want the spreadsheet to appear and typing OPTION-L. You then type D and the filename you wish to merge, ending with the "TXT" extender, followed by a return. While the sharing of information requires that you load different programs and save your data between them, it does let you do the same things that "integrated packages" let you do on computers with plenty of memory and big pricetags.

The SynCalc manual is excellent. It does a good job trying to cover all aspects of the program in a way that both novices and experienced computer users can understand and use. All of the documentation is typeset on slick paper put into a nine inch high, three-ring binder. (It seems that the nine inch binder is becoming the default size for all computer manuals these days.) The manual contains both a Table of Contents and a decent Index, which makes it easy to find answers to any question you might have regarding the program. It also comes with a handy reference card in case you want to be macho and dispense with the menus. The program and manual break with Atari tradition and acknowledge the author and Synapse. My compliments go to Mike Silva for doing such a good job on SynCalc. Atari has been criticized in the past for trying to keep their programmers anonymous on all but the Atari Program Exchange releases. The new applications programs, and other new Atari software give credit where credit is due.

All of the new Synapse programs are heavily protected and this can cause some problems with different disk drives. Owners of Happy enhanced drives will need to "slow down" their drives before they can load SynCalc. Off speed drives can also prevent the program from the loading. The manual has a section entitled "SynCalc Won't Load" that describes how to check for faults when the program won't load correctly. The indicate speed variations in drives may prevent SynCalc from loading, even when other programs work fine. I had intermittent trouble with an Indus drive running at 292-293 RPM. It had no trouble with an 810 drive running at 288 RPM. After adjusting the speed of the Indus drive to about 289 RPM, Syncalc loaded without problems. Anyone buying the new Synapse programs should check drive speed at the first sign of any problems loading the programs.

For those of you who say, "Well, this program is nice, but I run out of memory with a 48K machine in a hurry" these new programs may be for you SynCalc will let you take advantage of extra memory in your computer. It supports either the Axlon 128K Ramdisk or the Mosaic 64K boards. Other 64K boards work fine too, because the Intec 64K board in our 400 gave us extra memory to use. You can even use an 800 with a 64K board in each memory slot to get a total of 192K and SynCalc will access all of the memory. Marion and I are expanding our 800 with an Axlon Ramdisk to take advantage of this excellent feature.

So, how does SynCalc stack up? In the table at the end of this article, I have listed SynCalc's functions and features and compared it with Atari Visicalc and Lotus 1-2-3 for the IBM PC. Of course, since Lotus is an integrated package, the comparison isn't quite correct. However, I have listed the spreadsheet features in Lotus to give you an idea how close SynCalc comes to standard business packages. At $99 it is a must for anyone wanting to use a spreadsheet for more than casual home use. Besides, the authors of the new Snapse series have built ways into the programs so you can pass data between spreadsheets, databases, graphics, and word processing programs. It just isn't as easy as a single integrated program -- but it is much cheaper than buying an integrated package and an IBM PC. Now, if I can find the salesman at the downtown Kaufmann's computer department who laughed one day when I said I had an Atari....

Comparison of Spreadsheet Features

Feature/Command		SynCalc	Atari Visicalc	Lotus 1-2-3
Mathematical Functions				
@SUM()	sum of row or column	Y	Y	Y
@AVG()	average	Y	Y	Y
@ABS()	absolute value	Y	Y	Y
@ROUND()	rounds number	N	N	Y
@INT()	truncates to integer	Y	Y	Y
@RAND()	gives random number	N	N	Y
@SQRT()	square root	Y	Y	Y
@COS()		Y	Y	Y
@SIN()		Y	Y	Y
@TAN()		Y	Y	Y
@ATAN()		Y	Y	Y
@ASIN()		Y	Y	Y
@ACOS()		Y	Y	Y
@LOG()	base 10 logarithm	Y	Y	Y
@LN()	natural logarithm	Y	Y	Y
@EXP()	raise to power of e	Y	Y	Y
@PI()	gives value of PI	Y	Y	Y
Statistical Functions				
@COUNT()	counts items in list	Y	Y	Y
@STD()	standard deviation	Y	N	Y
@VAR()	sample variance	Y	N	Y
@MIN()		Y	Y	Y
@MAX()		Y	Y	Y
@MEAN()		Y	N	N
@RANGE()	range in row or column	Y	N	N
Misc. Functions				
@ERR	error display	N	Y	Y
@NA	not available display	N	Y	Y
@ISNA	data unavail. display	N	N	Y
@DATE()	gives serial date	N	N	Y
@TODAY()	date from log-in	N	N	Y
@IF()THEN()ELSE()		Y	N	Y
@CHOOSE()	choose an option	N	N	Y
@LOOKUP()	lookup in table	Y	Y	Y
Financial Commands				
@FV()	future value	Y	N	Y
@NPV()	net present value	Y	Y	Y
@PV()	present value of loan	N	N	Y
@PMT()	loan payment	Y	N	Y
@IRR()	internal rate of return	N	N	Y

Spreadsheet Comparison (continue

Format Commands			
Individual coulm widths	Y	N	Y
Center Labels	Y	N	Y
Left Justify Labels	Y	Y	Y
Right Justify Labels	Y	Y	Y
Fixed decimal points	Y	N	Y
Scintific Notation	Y	N	Y
Engineering Notation	Y	N	N
Currency	Y	Y	Y
Commas Inserted	Y	N	Y
Percent	Y	N	Y
- or () if negative	Y	N	Y
DATE (D-M-Y)	N	N	Y
DATE (D-M)	N	N	Y
DATE (M-Y)	N	N	Y
Worksheet Commands			
Protect cell contents	Y	N	Y
Windows: Horizontal/Vertical	Y	Y	Y
Synchronized Scroll	Y	Y	Y
Independent Scroll	Y	Y	Y
Fixed Titles: Horiz/Vert/Both	Y	Y	Y
Copy: Column or Row	Y	Y	Y
block of cells (row&column)	N	N	Y
Move: Column or Row	Y	Y	Y
block of cells (row&column)	Y	N	Y
Recalculate: Automatic/Manual	Y	Y	Y
By rows or columns	Y	Y	Y
Global Formatting	Y	Y	Y
Insert/Delete	Y	Y	Y
Print: to Printer	Y	N	Y
to File (text)	Y	Y	Y
range of cells	Y	Y	Y
cell formulas	Y	N	Y
Printer Codes in cells	Y	Y	N
Printer Codes in set-up	N	Y	Y
Macro commands	N	N	Y
DIF format save	Y	Y	Y
Convert from DIF format	Y	Y	Y
Convert from Visicalc format	Y	N/A	Y
Convert to Visicalc format	N	N/A	Y
Use Labels in formulas	Y	N	N
Sort (rows or columns)	Y	N	Y
Initialize a diskette	Y	Y	Y
Delete files on a diskette	Y	Y	Y
Goto a specific cell	Y	Y	Y
Allow label > column width	Y	N	Y
Memory Required	48K	48K	192K

SynFile+

BY SYNAPSE, DISK, $69.95

Reviewed by Joe Waters

Summary. SYNAPSE describes *SynFile+* as "the most powerful and advanced database management system ever created for ATARI Home Computers." This claim may, indeed, be true. But, before you run out and plop down your money, let me caution you that the first part of the phrase "most powerful and advanced database management system ever created" is qualified by the second part "for ATARI Home Computers." *SynFile+* does not compete with mainframe database management packages or with the more powerful 16-bit microcomputer packages like *dBASE III*. It is, however, functionally very competitive with any other product available for the ATARI, and, indeed, can compete favorably with many of the "filing manager" programs currently available for the IBM PC (excluding, of course, speed and data storage advantages provided by a 16-bit machine.) Bottom line: this is the system NOVATARI, and Current Notes, will use to maintain club records.

SynFile+ Packaging: IBM Compatible?. The first thing one notices about *SynFile+* is the packaging -- which is similar for all the SYNAPSE series. The program is delivered in a format which IBM has made popular: an attractive cardboard box encasing a three-ring notebook binder which holds the documentation and the program diskettes. The documentation is first class. It not only looks good, but is also well written. Two disks are included. One is the Program Disk, it contains all the code needed to run *SynFile+*. The second disk is a Tutorial, written in BASIC, that covers everything you need to know to use *SynFile+*. The binder, the documentation, the provision of the tutorial -- all are indications of a quality product. Only the flimsy nature of the outside cardboard box hints at the fact that this is a "poor man's" DBMS.

Learning SynFile+. The authors have structured the manual in the form of a tutorial that gently walks the user through the step-by-step creation of a demonstration database. In spite of the fact that the manual is quite good, most users will never read it. Most users never read any manual. The tutorial disk, however, will be used. Stick it in, turn on your ATARI, and there you have your own private course. An initial menu lets you choose between Database Introduction, Creating and Editing Forms, Entering and Retrieving Records, and Listing and Labeling Reports. As you work through the tutorial, the system simulates the operation of *SYNFILE+*. Instead of entering commands or data however, you press the space bar and the tutorial does all the typing for you.

SynFile+ is not difficult to learn. By the time you finish the tutorial, you'll be ready to go. After you create one, or perhaps, two databases, input some data and produce some reports, you will be an accomplished *SynFile+* user. When you learn how to create new databases from old descriptions and merge data from one file to another, you will be a *SynFile+* expert.

Learning *SynFile+* is certainly facilitated by the fact that *SynFile+* is a simple program to use. The system is menu driven: you're presented with various choices; use the cursor keys and select one. If that choice leads to another menu, it "pops-up" immediately and you make another selection. If the choice requires you to provide information, such as the number of characters to reserve for a particular item, you are prompted appropriately. If you don't like any of your choices and want to step back to the previous menu, simply press the ESC key.

I do want to leave the impression that *SynFile+* is easy to learn and easy to use. However, let me provide a small caution here. It's not perfect. ESC doesn't back you out of a choice in every instance. There are times when you have to throw in some garbage answers to get to a screen where ESC works. Similarly, there are other instances where I have run into little annoyances that I wish had been handled differently. I will point them out as I go along.

Using SynFile+. I want to give you a feeling of how *SynFile+* works by using it to create a simple database, for example, one that keeps track of your check-writing activities. Put in the Program Disk (no BASIC, *SynFile+* is written in FORTH), and turn on your ATARI. When you finally see the initial *SynFile+* screen (70 seconds), the bottom line of the screen presents you with three choices:

```
FILES     RECORDS     REPORTS
```

The MAIN MENU. There are only three initial choices. FILES deals with choosing the file you want to work with and includes many of the standard file handling commands (copy, rename, format, etc). RECORDS is used to maintain your database (insert, delete, and change records). REPORTS is used when you want to generate a list of information or preparing mailing labels.

When you start, the FILES option is highlighted. Press RETURN and a "pop-up menu" detailing the available FILE options appears:

```
Open          Subfile      Density
Close         Merge        Format
Copy          Rename       Delete
DIF->SynFile  SynFile->DIF
```

File Options. The OPEN option is highlighted. Before you press return once more, let's look briefly at the file options. The meaning of Copy, Rename, Delete, and Format should be familiar to anyone who has worked with disk files. Density. Yes, *SynFile+* does work on either single or double density disks (although if you have two or more drives, all must use the same density. ????? What a pity! I can't use an old 810 (SD) with a new INDUS.(DD)

The "DIF" (Data Interchange Format) options are used to send data to another program (SYNFILE->DIF) or to input data from another program (DIF->SYNFILE). *SynFile+* allows transfer of data. How fast is another matter. I wrote a program to convert a FileManager 800 database of 150 records (with twelve fields) to DIF format. I then used

the *SynFile+* option to convert this DIF file into a *SynFile+* file. Although it worked, it took one hour to complete the conversion. In the time it took to write my conversion program and then run it successfully, I may very well have been able to get the task done quicker by just keeying in all the data.

(Note: I had planned to include my conversion program in *Current Notes*. However, I discovered that SYNAPSE will have a conversion utility to move *FileManager 800* files directly to *SynFile+* files by the time you read this. They will make that disk available free to ATARI User Groups.)

Subfile is used to create a subset of an existing database and Merge is used to combine one database with another (similarly structured) database. Before anything can be done to a database, it must be "opened"; you would use the OPEN option. When you are done using a database, you use the CLOSE option.

Creating a Database. The first step in creating a database, therefore, would be to use the OPEN option under FILES. Press RETURN and, normally, a list of all the files on your data disk appears. Initially, there are no files on your data disk. In fact, the Program Disk is still in the disk drive. The only thing that appears on the screen is:

```
* CREATE *
```

If you had placed a data disk in the drive before you tried the OPEN option, all the data files on your disk, as well as the CREATE option, would have been listed. You would move the cursor to the file you wanted to open and press RETURN. To create a new file, you choose the CREATE option.

Unfortunately, the CREATE PROGRAM is not in the main SYNFILE program. Therefore, if you have a data disk in the drive and you want to create a new file, you have to remove the data disk and insert the Program Disk to load Create. Once it is loaded, you remove the Program Disk and put your data disk back in. After you "create" the structure for your new database, you save it to your data disk, remove that disk, put in the Program Disk, and reload *SynFile+*. All the other features in *SynFile+* are in the main program. If only SYNAPSE could have fit CREATE in as well, the user could have avoided all this disk swapping. Unfortunately, 48K does have its limits.

Why not put the data disk in drive 2 and keep the program disk in drive 1? Another "unfortunately." *SynFile+* will only open data disks on drive number 1. This is one of those little annoyances.

The Create Menu. Once CREATE is loaded, another sub-menu appears above the FILES option:

```
Create form   Edit form   SynFile+
```

The names of all files on your data disk are listed above the menu (in this case nothing is listed since nothing is there). Create form is highlighted. Press RETURN and a prompt appears in the menu window asking you to supply a name for your database. Enter a name and press RETURN. Now you are greeted with an essentially blank screen. It's time to define the form you would like your database to take.

Designing a Database Form. Imagine your database as a sheet of paper with one row of information for every check you write. You have six columns labeled "Check No.", "Date", "To:", "Amount", "Item", and "Code." Everytime you write a check you would fill in another row on your paper. Each row represents a separate RECORD. The column headings are called FIELDS. Each field represents some item of information, pertaining to a specific check, that you wish to record. Some items are dollar amounts (Amount), some are textual information (To, Item, Code), some represent integer numbers (Check No.), and some may be dates (Date). The first step in creating a database is nothing more than specifying the number and type of items (FIELDS) we want to keep in our database. This is what the CREATE option does in *SynFile+*.

Positioning the Fields. A form is used to define a database. You start with a blank screen of 21 rows and 80 columns (the screen scrolls left or right as needed showing you 40 of the possible 80 columns at any one time). Use the cursor keys to move wherever you want the first item in your database to be displayed on this form. When you are at the position you want, type in a name for this first field (*SynFile+* allows up to 64 fields). The name can be anything you want (up to 31 characters) and will be the name you use to refer to this item later for creating reports. Press RETURN and another sub-menu appears:

```
Text          Numeric      Cumulative
Look-Up       Dollar       Record #
Date          Integer      Counter
Conditional   Computed
```

SynFile+ Field Types. You may use any of eleven different field types to characterize your data. The meaning of Text is obvious. SynFile+ allows up to 255 characters in a text field. That's over three lines on an 80-column page. Plenty of space if you'd like to keep free-form comments in your records.

Numbers can be stored as dollar amounts (Dollar), integers (Integer), decimal numbers (Numeric), or computed values (Computed). This last type lets you define one item in your database as a function of other items. The definition can include the usual arithmetic operators (+ - * /) as well as six *SynFile+* functions (ABS, SQRT, LOG, LOG10, EXP, and EXP10).

The Conditional field lets you define a text field as being conditional on another field in the form. For example, you might have a field called BALANCE which could display as either "Profit" or "Loss" depending on whether the value in some other field was greater than or less than zero.

The **Date** field type will automatically format as __/__/__ to hold the month, day, and year. In addition, when you are entering data, the contents of the date field will remain constant with each new record as new records are entered. Thus, you can enter today's date on the first record you insert and that date will automatically be carried forward for every record you enter until you explicitly change it.

The **Look-Up** field can be particularly useful. It represents a text field where you specify, in advance, all of the allowable entries. For example, if you are recording student schedules, the number of possible courses is fixed and known beforehand. You could record all the course titles in a look-up table. When records are being inserted, only values in the look-up table would be accepted for this field. In fact, if you press the ATARI key (normally gives you inverse video), the permissible values for the look-up field are automatically displayed. When you see the one you want, hit **RETURN** and it is entered! If you do not enter anything into this type of field, the first value in the look-up table (the default) is automatically entered.

The **Record #** and the **Counter** field types are also useful. With **Record #**, *SynFile+* will automatically insert an integer (starting with 1 and increasing by 1 up to a maximum of 32,767) for every new record you insert into the database. The **Counter** field type allows you to specify the starting number (between 0 and 999) and the increment (1-100).

A completed input form for our checkbook database might take this appearance:

```
CHECK NO.: _______  DATE: __/__/__

TO: ------------------------------

AMOUNT:$__________  CODE: ___

ITEM: ----------------------------
```

Changing Your Mind. A very useful feature of forms design in *SynFile+* is that you are not stuck with your initial layout. If you want to move any particular item, just move the cursor to the first letter in the item name and press RETURN. Another sub-menu pops up. This one lets you either **MOVE** the item, give it a **NEW NAME**, a **NEW TYPE**, a **NEW LENGTH**, or just **DELETE** it. If you choose to move it, the item name and it's field are changed to inverse video. You use the cursor keys to move this field. When you are happy with your new position, simply press **RETURN** and the field is moved permanently.

Note: even after you have created your form and inserted records into your database, you can change the form by using the **EDIT FORM** option. Moving a field or changing its name or changing the length of a numeric field can be accomplished without making any changes in the database. Other changes, such as changing field length or changing the type of field will require you to save your "revised" form under a new database name. The information in your original database can then be transferred to the revised database by using the *SynFile+* **MERGE** option.

Indexing Your Database. When you return to *SynFile+* and open your newly created database, the first thing *SynFile+* requires is that you choose one or more indexes. *SynFile+* keeps track of your records by creating an index on one (or more up to a maximum of 16) of your fields. If you try and retrieve a record based on the contents of a field in your index, the retrieval is very quick -- *SynFile+* knows exactly where the record is stored on disk. If you retrieve a record based on a non-indexed field, *SynFile+* has to search through every record in your database to determine whether the record meets the search criteria or not. Index on the fields you will later be searching on.

When you list your data, it will appear sorted (ascending or descending, whichever you prefer) on the index field(s). If you want your data sorted by some other field, it is a simple matter to re-index the file on whatever new field(s) you want.

Entering Records. Entering data is simplicity itself. Open the appropriate file, move to the RECORDS option, and press RETURN. You will see the following sub-menu:

```
Retrieve    Update all    Re-index
Enter       Delete all

FILES      RECORDS      REPORTS
```

Select **Enter** and the form you just finished designing appears on the screen. Fill it in. You can use the cursor control keys to jump from field to field making whatever changes you like until you are satisfied you have the correct data entered. Press START and the record is automatically inserted into your database and a new blank form appears. Enter as many records as you like.

Retrieving a Record. Choose the **Retrieve** option and, once again, your blank form appears. You specify the records to retrieve using this blank form. For example, if you wanted to recall check number 123, you would enter "123" in the "CHECK NO." field. Only the record that had a value of "123" under check number would be retrieved. If you wanted to retrieve all checks that were greater than $100, you could move to the AMOUNT field, press the ")" symbol and then enter then number 100; only checks with amounts greater than (or equal to) $100 would be retrieved.

Search Criteria. If you want an exact match, enter the match you want in the appropriate field. You can also request items less than or equal to "\", greater than or equal to ")", or not equal to "#". Note: you cannot specify a range as a search criteria, e.g. "greater than 50 but less than 100".

On text fields you can use the asterisk (*) as a wild character. Thus, "A*" would mean anything that begins

with the letter A, "*MD." would mean anything that ends with the letters "MD.", "*ATARI*" would mean anything that has the letters "ATARI" anywhere in the text field. Ranges are similarly not available in text field searches, e.g. you can not request items "greater than B and less than D."

You can search on two fields at once, for example, find all checks > $100 AND code = "CL". In fact, *SynFile+* allows you to search on up to 16 different fields at once. When multiple fields are used, you specify whether to use AND or OR in satisfying the search criteria.

Updating a Record. When you retrieve a record, the form you designed is displayed on the screen with the data relevant to that record. "Update" is the default mode. If you want to make a change, move the cursor to the appropriate field and change it. When you go on to the next record, the changes you made are automatically stored on the diskette.

A particularly nice feature of *SynFile+* is the multiple update capability. Suppose you had recorded LWC in the "To" field on a number of different checks. Now you want to go through your database and change "LWC" to be "Local Water Company". You would use the UPDATE-ALL option. Enter a search criteria that would identify all records with a "To" field value of "LWC". Another blank form would then appear in which you would enter any changes you wanted to make. Enter the new expression in the "To" field. Press RETURN and all the relevant records in the database are updated.

A multiple delete capability is also included. If you specify a search criteria "Date" < "12/30/84", every entry with a date prior to 1985 would be deleted. This could be an incredible time saving feature. Of course, if you make a mistake and forget to put in any search criteria, every record in your database is deleted. But since you made a back-up of your data disk, there is no real harm done. You do make back-ups, don't you?

Reports in SynFile+. I'd love to be able to say that *SynFile+* gives you any kind of report you'd like to have. But flexibility in report formatting is one of the features that comes only with the higher-priced DBMSs. There are only three different ways to look at your data using *SynFile+*. You can examine (and print) an individual record, you can print a "list" where every line corresponds to a record, or you can make up "labels" where the positioning of items is up to you.

Individual Records. The printed version of your individual records will look exactly like the form you designed. Field names as well as the data in the field are printed on paper in the precise locations you designed into your form. To print a record, you first retrieve it. When it is displayed on your terminal screen, press the OPTION key and a sub-menu appears offering you a choice of printing the record, calculating the record, or deleting it. If you choose PRINT, the record is printed on your printer. (The calculate option is used to force *SynFile+* to calculate "computed" fields and allow you to view the results before the record is saved.)

Printing a List. Both the list and label reports are found under the REPORTS option of the Main Menu. To produce a list, you enter the name of the field you want listed on a "print format line." You may include up to 40 field names (columns) up to a total of 232 characters. The field names become the column headings. If you append a "+" to numeric field names, *SynFile+* will calculate a column-total and print it at the bottom of your list.

SynFile+ will automatically determine the amount of spaced needed for each field (the greater of the field length or field name). However, you can change this default setting by repositioning the field name on the print format line. For example, if you wanted to leave an additional five spaces between the first and second fields in your list, you would move the cursor to the beginning of the second field name and use the CNTRL - INSERT key combination to insert five blanks.

You can send your list (or labels) to the screen, to a disk, or to the printer. If you send your list to the screen, however, you will only be able to see the first 40 columns. The screen does not scroll to allow you to see any data beyond column 40.

If you choose to send the list to printer or disk, you can specify the total page length and include a report title. The page length feature will print the number of lines you request and then eject to a new page, print the column headings again, and continue on with your list. The report title only appears at the beginning of your list and is not repeated on every page. You can use the report title to send any desired printer codes (for example, to turn on condensed print) to your printer.

The Labels Option. When you select the labels option, *SynFile+* gives you a screen very similar to your initial create form screen. Move the cursor anywhere on the screen and position fields wherever you like. This function is used primarily for generating mailing labels but can have other uses as well. You are allowed an arrangement of fields on an 80 column by 21 line form. The left margin, lines between labels, and the column between labels can all be set from 0 to 999. Up to 32 such forms can be printed across the page.

SynFile+ allows up to 32 fields to be displayed in a label. A trailing ";" concatenates adjacent field data and prints a comma and a space. A trailing "&" concatenates adjacent field data and puts a space between the data.

After you define your label placement, and indicate the number of labels on a page and the desired column and line spacing, you are allowed to indicate any desired search criteria. Finally, you select **PRINTER**, **DISK**, or **SCREEN** to receive the output. You come back to this choice after your report is finished in case you want to send it to another device.

If your initial label design was not quite correct, that's too bad. Because *SynFile+* does not remember what you specified for your list or label entries. If you g

back, the print format line for lists and the screen for labels is blank. This is another major annoyance. You seldom get things right the first time. If *SynFile+* remembered your last selections, all you would have to do is edit it to make any desired changes.

SynFile+ Utilities. This review has already gone longer than I would have liked. So I will not dwell on the various utilities. I will say that it is possible to use the DIF option to move data from other systems into SynFile. The data comes in with fields named "A", "B", "C", etc. and with a minimum text field length of 16 characters after conversion. You can use the edit form option to change field names. If you want to change field lengths as well, you have to define a new database with the same names as that in your original DIF conversion and then use the MERGE option to combine the two. Although it all can be done, it will take a good deal of time and lots of patience.

You send *SynFile+* reports to *ATARIWRITER* by sending your report to a disk file and then later using the merge option in *ATARIWRITER* to include that disk file in your *ATARIWRITER* program. I have not tried this option.

Conclusion. I have covered the use of *SynFile+* in great detail. If you were already familiar with File Manager 800 or Data Perfect, this review should give you a good idea of the relative merits of *SynFile+*. If you never used any DBMS, you should by now have a pretty good idea of just what one does. Is *SynFile+* worth it? To some people, YES; to others, NO. I do know that with the recent reduction in prices, the proportion of people who will conclude "YES" will certainly increase substantially.

SYNCALC

BY SYNAPSE, DISK, $69.95

Reviewed by Jack Holtzhauer

SynCalc, one of several integrated applications programs recently developed for the ATARI by SYNAPSE, is an "advanced electronic spreadsheet". So what's an electronic spreadsheet and why do I need one?

Electronic Spreadsheets. Very simplistically, I guess a spreadsheet might be compared to a large piece of graph paper, each square (or cell) of which can contain text, numerical data, or a formula used to calculate a numerical result. The 128 columns on the *SynCalc* spreadsheet are identified by letters running from A to DX. The rows are identified numerically and run from 1 to 255, providing a matrix of 32,640 cells. Individual cells are identified by their column/row coordinates. The cell at the intersection of column A and row 1 is cell A1. The cell at the intersection of column W and row 42 is cell W42, ad infinitum.

CURRENT NOTES

A simple use of a spreadsheet might be to calculate the area of a circle by inputing the radius (R) in cell A1 and the formula P1*(A1)^2), or P1*R², in cell A2. Upon calculation, the area of a circle with the given radius would appear in cell A2. Admittedly, not many of us would use a spreadsheet for such a purpose, but suppose this was just the first step in a series of complex engineering calculations you wished to perform for a wide range of radii. By merely entering your complete series of formulae on the spreadsheet, you can perform your repetitive calculations by merely changing the single value in cell A1.

Personally, I have found spreadsheets to be an invaluable aid in handling family finances. Several days after I retired back in '79, my wife stormed out of the house muttering the old gag line "for better or for worse, but, damnit - not for lunch!" I had been under her feet "systemizing" her kitchen by alphabetizing all the little spice cans, etc. To keep me at arm's length, she promptly got a job selling new homes for a major builder. She now earns both salary and commission income, part of which is invested in company stock, employer sponsored mutual funds, and a couple of tax dodges. Projecting tax liability with all the employee expense deductions, auto costs, 401K income exclusions, investment fund earnings, capital gains (and losses), etc., was a real hassle. I solved the problem by dumping all the applicable IRS forms onto *SynCalc* spreadsheets. Now, once a month, I merely enter our newly projected annual income and related data on the applicable sheets, and all the calculations are updated for me.

In short, to quote from the first page of its manual, *SynCalc* ". . . can function like a business or scientific worksheet, combining the convenience of a pocket calculator with the powerful memory and electronic screen capabilities of the personal computer." As illustrated in the simple example given above, it's great for "what if" types of forecasting or planning analysis and there are a number of books on the market describing a wide variety of possible ways by which the home computerist can make use of a spreadsheet program. More on these later, but first - - -

Documentation. The one-disk *SynCalc* program comes packaged in a 9x7 three-ring vinyl covered looseleaf binder with a rigid cardboard dust cover. The documentation, consisting of over 140 pages, includes a folding command guide which can be removed from the binder for ready reference. The instruction manual itself is divided into three parts (no intent to plagarize the first sentence from Ceasar's Gallic Wars). A general introduction is followed by a two-part tutorial which steps the new user thru the intricacies of the program. The final section, some fifty pages in length, provides a detailed description of *SynCalc* commands, functions, error messages, etc., and offers a few hints on saving space and improving speed of calculation. An index is also included.

Hardware. System requirements include 48K and at least one disk drive (will support two). If booted on a non-XL ATARI, you'll have 25% free RAM for program use, but only

21K if using an XL (increased to 25K if Translator Disk used). The program will also recognize the presence of an Axlon Rampower 128K or Mosaic 64k Select board, giving you the extra RAM to play with. (Available at L&Y Electronics, Woodbridge, VA., at $49.95 with an additional 5% discount for users' group members presenting ID cards. Incidently, John Linton of L&Y reports SYNAPSE will provide a back-up disk for $20.00 upon receipt of your warranty card and proof of purchase.)

A couple of major features of the *SynCalc*, *SynFile+*, and *SynTread/Graph* programs before we go any further. All three can produce data interchange format files (DIF files) for use by its sister programs. In addition, *SynCalc* and *SynFile+* can produce text files for use with *ATARIWRITER*. This mutual compatibility provides the user with a fairly powerful package of integrated word-processing, spreadsheet, database, statistical and graphing applications programs.

SynCalc vs VisiCalc. The only other major spreadsheet program currently available for the ATARI is *VisiCalc*, and it might be best to describe *SynCalc* with this in mind. How does it compare to the ATARI version of the *VisiCalc*, the godfather of all spreadsheets? Very well, indeed. It has all the features of my first-generation ATARI *VisiCalc*, with some added ingredients. *SynCalc* includes the statistical functions sample mean, sample standard deviation, sample variance and range (difference between the minimum and maximum values within a range), all features missing from *VisiCalc* for the ATARI. Unlike *VisiCalc*, *SynCalc* also includes the logical function IF/THEN/ELSE and the financial functions needed to compute the future value of an annuity or a loan payment, given the principal, interest and term of the loan. *SynCalc* will even sort alpha or numeric data for you and, if you wish, duplicate it in its new configuration elsewhere on the sheet, or merely replace it from whence it came. Text entry is also easier. *VisiCalc* permits text entry only cell by cell. If you're in a six space cell and wish to use a nine letter heading (label), you've got to toggle to the next cell to enter the last three characters. *SynCalc* permits text entry to overflow to the next cell, and the next, as necessary. The list goes on and on! Is Synapse confident their product beats the ATARI version of *VisiCalc*? Could be. They've included a routine for those of you with *VisiCalc* files you'd like to convert to *SynCalc* format.

SynCalc is compatible with most printers and, if desired, will generate condensed or compressed type with the ATARI 825 & 1025, EPSON, GEMINI and MEC/C.ITOH products.

Working with SynCalc. Mastering program execution is not difficult. The user has three choices as to how he might wish to execute most formatting controls or "DOS" functions. Two involve use of the menu tree. The third is the "expert user's mode". For example, if you wish to change the width of column I to twelve spaces from the default value of eight, you can use the menu tree to accomplish your purpose. You first hit the OPTION key to display the four sub-menu selections on the bottom of your screen, then the arrow keys and return to make your selection - COMMAND in this instance. The 16-item COMMAND sub-menu is then displayed. You again use your arrow keys to toggle to WIDTH and hit return. You are then asked how many spaces you wish in the column in question, 12 in this instance. After responding to this query, you're asked to identify the first column in the range of columns you wish to change. You can then merely enter the column letter, I in this instance (menu/keyboard entry mode), or toggle your screen cursor to column I and hit return (menu/cursor point mode). You identify the last column in the range in a similar fashion, and your mission is accomplished.

Does that sound complex and time consuming? Maybe so. But you can accomplish the same purpose with only six keystrokes using the expert user's mode - "/FW12I", slash for expert user's mode, F for format, W for width, 12 for value and I for column involved. Incidently, as you toggle through either of the menu command modes, the appropriate expert user's command is displayed in the upper left-hand corner of the screen - a feature most helpful to the new user.

Function Summary. Inasmuch as this article is meant to be an overview, not a tutorial, I'm going to close by merely providing you with a listing of most features, functions, and formating controls *SynCalc* makes available to the user. (See accompanying table.) Readers generally familiar with spreadsheet programs may find this helpful in analyzing the completeness of the package.

References. But others may still be wondering whether a spreadsheet program might be worth an outlay of fifty-bucks. If so, your local public library is probably stocked with several of the scores of books written on this subject. One such tome, written by Donald H. Bell and published by RESTON ($14.95), is *The VisiCalc Book, Atari Edition* - - - an in-depth tutorial on the use of *VisiCalc*. It includes a sixteen-page bibliography on spreadsheet publications. Stanley M. Trost's *Doing Business with VisiCalc*, published by SYBEX at $11.95, offers a compendium of uses for spreadsheet programs ranging from a simple checkbook register to a seasonal forecast of retail sales. *VisiCalc Home and Office Companion*, written by Castleuitz, Chisausky and Kronberg and published by OSBORNE/MCGRAW-HILL at $12.95, offers templates on such varied spreadsheet uses as cash flow analysis and EEO personnel reports. All of these publications can be easily adapted to SynCalc use.

Your needs may be simpler or much more complex than mine. But why not give it a try. What's fifty-bucks. Who are we kiddin' anyway. We all need somethin' to keep us off the street. And aren't our little computers just like electric trains - toys for big boys!

Table 1:
SynCalc Features, Functions, and Formatting Controls

DOS FUNCTIONS

LOAD..... worksheet or data (DIF)
SAVE..... worksheet, data or text (ATARIWRITER)
FORMAT DISK
DELETE DISK FILE
RENAME DISK FILE
CONVERT VISICAL TO SYNCALC

FORMATTING CONTROLS

Formatting controls may be used globally, or confined to a specific cell or series of cells:

NUMERICAL AND TEXT

JUSTIFY... left, right or centered
MARGIN... fix margin at left of cell
COLUMN WIDTH can be individually set from 1 to 36 characters
PROTECT.. protect cell or range of cell from further formatting commands or recalculations
UNPROTECT reverses action of above command
UNFORMAT delete formating controls from cell or range of cells

NUMERICAL

Numerical data may be presented in four general formats - FIXED (integer), FLOATING, SCIENTIFIC or ENGINEERING. In addition, FIXED and FLOATING numbers may be further formatted by:

SETTING SPACES TO RIGHT OF DECIMAL POINT
PRECEDE WITH DOLLAR SIGN
INCLUDE COMMAS TO LEFT OF DECIMAL POINT
TRAILING PERCENT SIGN
EXPRESS NEGATIVES as: -, ( ), OR CR

MATHEMATICAL FUNCTIONS

ABS	ATAN	INT	PI
ACOS	COS	LN	SIN
ASIN	EXP	LOG	SQRT
			TAN

STATISTICAL FUNCTIONS

AVG..... averages a range of values
CNT..... counts the number of numeric entries within a range
MAX..... maximum value
MEAN..... sample mean
MIN..... minimum value
RNG..... range (between min & max)
SD..... sample standard deviation
SUM..... sum of values
VAR..... sample variance

FINANCIAL FUNCTIONS

FV..... future value of an annuity
NPV..... net present value of a range of investments
PMT..... loan payment

CELL MANIPULATION

COPY..... copy contents, format and formulas of a cell or range of cells to another cell or range of cells.
MOVE..... cell or block of cells to another position
INSERT... row or column
DELETE... row or column
ERASE.... cell or range of cells

MODE OF CALCULATION

Calculation may be set for either column or row priority and, in addition, may be either:

AUTOMATIC recalculates after each entry, except PROTECTED cells
MANUAL... recalculates by pressing START key, except PROTECTED cells
FORCED... recalculates all cells, including PROTECTED cells

SCREEN PRESENTATION

WINDOWS.. vertical or horizontal with synchronized or independent scrolling
TITLES... fix vertical
fix horizontal
fix both
HEADINGS toggle on or off

MISCELLANEOUS FUNCTIONS

LOOKUP... allows user to set up table of values to determine a second value (e.g., tax table)
SORT..... sort rows in selected range in either ascending or descending order based on selected sort column, and place at selected position on sheet
PRINT.... dump any portion of sheet to most printers; compressed font on ATARI 825/1025, EPSON, GEMINI and NEC/C.ITOH

COMMAND WILD CARDS

@..... entire worksheet
:..... bottom-right cell
;..... current cursor position

SynFile+: A Review

by Rick Giarl

In the 2 years or so that I have owned my Atari 800 I have been asked many times what kind of computer I owned and what exactly do I do on it. It seems that about half the time, after I say I have an Atari, the person I am talking to says "Oh, so you play a lot of games, huh??" Well of course I play some games, heck even the people who use all those other computers play some games on them. But it seems that most people, mainly those unfamiliar with the Atari, know it only by reputation as a game playing machine. And many "serious" computer users look down on it as not being a "real" computer.

Well lets face it the Atari IS possibly the best game machine on the market. It's graphics and sound capabilities are unmatched by anything even near it's price. And there are many, MANY simply great games out for it. But the heart of the Atari computer is the 6502 microprocessor the same basic chip that is the heart of other "more serious" computers, such as the Apple II series. There is no real reason why the Atari could not do most of the things other computers do. Things like Word Processing, Spreadsheets, Data Base Management, and other Business programming. Someone must simply sit down and write the proper programming for these applications.

And that seems to be part of the reason why outsiders think of the Atari as simply a game machine many of these other programs are simply not being written. Or at least most of the programs being written for the Atari are games, with the few "Business" programs being written either not up to par with "serious" business programs, or not getting the attention and advertising they deserved. There are several business-type programs out there for the Atari that are simply great -- but for some reason or other we do not hear much of them. Maybe the authors/publishers of other programs do not think that a "game machine" can generate the market demand for them to spend the money it takes to bring their products to the marketplace. For those reasons, and perhaps others, we Atari owners are getting the short end of the stick when it comes to "serious" computing.

Continued on page 13

Continued from page 12

Or, at least, that has been the case. Last year a new series of programs for the Atari was introduced. Although it has taken some longer than expected time to get these programs into the stores, the Syn series from Synapse is now available. With these programs available, we are now a leg up on the rest of the home computer market. This series is made up of 3 major programs along with a few somewhat simpler additional offerings. The 3 main packages go by the names of SynFile+, SynCalc, and SynTrend. SynCalc is a Spreadsheet program (like Visicalc plus a lot more) and SynTrend is a visual Graphics and Plotting program (like B-Graph). These programs have been or will be reviewed elsewhere. This article will focus on SynFile+.

Synfile+ is a general database generator and maintainer. It allows the user to define a database to his own specifications and then maintain and use that data however he wishes. He can enter records, update them, delete them, or manipulate them in many different ways. "But", you ask, "what can I use it for?? I don't own or run a business and I can't see a use for a program like that!" Well, there are many uses for such a program aside from running a business. First of all you must know what a datbase is: a data base (or database, whichever) is simply any organized collection of facts. It does not matter what those facts are or how/why they are organized. Databases exist in many shapes and forms all around us everyday. Possibly the best known and most used database of all is the telephone book. It contains the names and phone numbers (facts) of the people who have phones in a certain geographical area, arranged in a certain order (usually alphabetical). Other databases are things like recipe books or files, almanacs and encyclopedias, and the card files at your local library. Even your checkbook can be considered a database!

SynFile+ can do all of those things and more. Billed as "The Ultimate Filing System", it has more features than any other data base program for the Atari computer and, I am told, seriously rivals other data base programs such as Dbase II. I am also told that it contains many functions usually found only on larger mainframe databases. Whether this is true or not, it is certain that SynFile+is a great tool for the Atari. So what makes SynFile+ so great? Well, from my own point of view one of the biggest advantages of SynFile+ over other Atari data programs such as Data Perfect and Filemanager 800+ is its capacity. On a normal 48K Atari system you can maintain data bases containing up to about 2050 records with SynFile+, compared to about 730 records maximum on Filemanager 800+ and a similar amount on Data Perfect. Although the maximum capacity for Data Perfect depends on how big each record is - actually, with very small records, Data Perfect can hold about as much as SynFile+. SynFile+, however, is not limited to small records to achieve a large capacity. Data Perfect maintains it's entire data base in memory and only saves the data base to disk when you quit each session, it is effectively limited to 48K worth of program and data. The longer the length of each record the smaller the number of records possible. Both Filemanager 800+ and SynFile+ use a different method to keep track of records. What they do is to maintain the data on disk while keeping an INDEX of the data in memory. This index is kept sorted and is basically the pointer information necessary to tell the disk drive exactly where to go to access any particular record. The biggest disadvantage to this system is speed; the program must locate in memory the index needed and then input the record from the disk drive. With the comparatively slow Atari disk system, this means some delay in accessing any particular record. The biggest advantage is that the capacity of the system is not quite as dependent on the length of the data record as is Data Perfect. This system of maintaining the index in memory is, however, about where the similarity between SynFile+ and Filemanager 800+ ends. Whereas Data Perfect is in machine language, Filemanager runs on Atari Basic. SynFile+, however, is written in Forth. This gives it nearly the speed of machine language, and eliminates the need for the Basic cartridge, freeing additional memory for it's use. In addition, SynFile+ is not limited to a single single-density disk as is Filemanager 800+. SynFile+ supports single, enhanced (1050) or true double density drives. In addition, with SynFile+ your data base can stretch over up to 16 disks. While keeping track of which disk is which can get a little hairy, if you need it, the ability is there.

SynFile+ also has one other ability that the others don't have: it supports both the Axlon RamdisK and the Mosaic 64 Select boards. Yes, I know, Filemanager 800+ also can use the Axlon RamdisK. But what Filemanager 800+ does is to load your data base from the disk drive into your RamdisK and then

Continued on page 14

Continued from page 13

use the Ramdisk for faster access. While this is great for access speed, you are still limited to 88K worth of data storage (the size of a single density disk). SynFile+ uses this additional memory to maintain the index of your records. When combined with the ability to stretch files over 16 disks, this means LOTS of records possible. While it will support the Axlon Ramdisk, it will also support 1, 2, or 3 Mosaic 64K Select boards. To give you some idea what this means in relation to capacity -- with 3 Mosaic 64K Select boards in an Atari 800 (192K) you can maintain a database of up to 34,813 records, depending on how you index it (The figure of 34,813 was obtained using a look-up field as the index). That is probably many more records than 99.99% of the users will ever need!!

In addition to the capacity in terms of the number of records, the program also has a lot going for it so far as it's general capabilities are concerned. For instance, each record can contain up to 66 fields. And the field names can be up to 31 characters in length. The field names can use all valid ASCII characters except "(" and ")". Your created data base is "free form" in that each field can be positioned, and moved if desired, whever wanted. And it is not limited to the size of your TV or monitor screen: the form size is 80 characters by 21 lines inclusive of field names. With SynFile+ you scroll a 40 column window over 80 columns. Unlike Filemanager 800+, though, it is limited to a single 80x21 page.

Another strong point for SynFile+ is the large number of field types allowed. These types include the following:

1. TEXT -- up to 255 characters in length.

2. NUMERIC -- fixed or floating point ranging from -9x10(97) to 9x10(97).

3. CUMULATIVE -- to keep a running total of values contained in other fields.

4. LOOK-UP -- This field will contain an entry from a table of text data, limited to 34 characters and automatically right justified.

5. DOLLAR -- Automatically right justified with a $ sign and decimal point. Up to $999,999,999.00 without exponential notation. More with.

6. RECORD # -- an automatic ascending sequential number supplied by the computer, starting with 1 and goeing up to 32767 in increments of 1 with each new record.

7. DATE -- month, day and year in format mm/dd/yy. System supplies slashes.

8. INTEGER -- whole numbers ranging from -32768 to +32768.

9. COUNTER -- same as record number except that user supplies the starting number (0-999) and increment (1-100).

10. CONDITIONAL -- designates that the field will contain a particular text entry depending on a conditional relationship defined up to 34 characters long.

11 COMPUTED -- Based on formula you define. Supports +,-,*,/, ABS, SQRT, LOG, LOG10, EXP, and EXP10.

Even with this large number and variety of field types, SynFile+ is actually very easy to use. Like Filemanager 800+, and unlike Data Perfect, it is menu driven. Unlike Filemanager, howeverm the menus on SynFile+ do not cover up your data by taking up the entire screen. Usually in fact the menus take up only the bottom couple of rows, leaving the rest of the screen visible. For instance, when first booting up the program all that shows is 1 line accross the bottom of your screen -- the main menu. By using the right and left cursor keys you move the "highlight" to whichever of the 3 main menu choices you wish to use: FILES, RECORDS, or REPORTS. Then simply press the [RETURN] key. When initially booting up, your choice will be "FILES" as you must load a file before you can work on it.

Immediately after pressing the [RETURN] key you will see the Files Sub-menu. This sub-menu contains many options including:

OPEN	SUBFILE	DENSITY
CLOSE	MERGE	FORMAT
COPY	RENAME	DELETE
DIF-Syn	Syn-DIF	

Continued on page 15

Continued from page 14

The first option, OPEN, will automatically be highlighted when the sub-menu comes up. By simply pressing the [RETURN] key at this point the program will look out to disk drive #1 to see what SynFile+ files are there available to be loaded. This brings up a point worth mentioning. Unlike Filemanager 800+, SynFile+ is a single load for normal usage: once loaded the SynFile+ disk is not needed, leaving the disk drive free for use by the data disks. The system works very well; you don't have to worry about "swapping" disks like you do with a single drive and Filemanager. There is one exception to this -- the "CREATE" module, needed to create or modify a data file form, must be loaded seperately when it is needed. In other words, if you wish to create or modify a form you must be prepared to have to swap your SynFile+ disk and your data disk once or twice. But other than for that, once the program is in your computer it does not need to go back to the program disk.

With a data disk in drive 1, when you highlight the "OPEN" option and push [RETURN] you will be shown a listing of all SynFile+ files on your data disk. Once again, by simply using the cursor control keys you select which you wish to load, or specify "CREATE". The essential parts of your form will be loaded into the computer and set up, along with the index of your data. When the load is completed, displayed accross the top several rows will be information on the loaded file. This includes the datafile name, the index length, the number of records the file contains, and the capacity of the file (based on memory size and the index length). Also showing will be the indexed fields in the order in which they are indexed as well as their index length and the main menu. At this point, with a file loaded, it is time to do something with it. If we change the main menu to "RECORDS" and press the [RETURN] key we will get the "RECORDS" sub menu. This menu includes the following options:

RETRIEVE UPDATE ALL
RE-INDEX ENTER
DELETE ALL

Once again, by using the cursor control keys and the [RETURN] key we can select the operation we wish to perform. As "RETRIEVE" will probably be the most used function, it is the one automatically highlighted.

I don't want to take the time to go into detail about these options, but I do want to mention 3 things at this time. First, when using "RETRIEVE" you will be asked to specify search criteria. By simply pressing the START button you specify ALL records. And to get each subsequent record, you again press the START button. Second, SynFile+ will automatically position the cursor at the start of the next field needing data. It starts at the upper left corner and proceeds left to right, top to bottom. Whenever you press the [RETURN] key, whether or not you have actually entered any data into that field, it will jump to the next field. If it is at the last field, it will jump back up to the first field. And it moves FAST. Third, whenever you are in the "ENTER" or "RETRIEVE" modes, if you press the OPTION button a new menu comes up allowing you to PRINT, CALCULATE, or DELETE the record. If PRINT is selected, the entire record exactly as shown on the screen will be printed out to your printer, including all field names.

There is still so much more to mention about this program!! But not enough time to do it, so I will merely touch on some of the highlights. For instance, you can do searches on multiple fields with the options AND or OR. You can also use wildcards for text searches, as well "greater than", "less than", or "not equal to". When doing reports you can use either a "LISTS" option or a "LABELS" option and can output to a printer, a disk, or the screen. When using the "LISTS" option you use a PRINT FORMAT LINE to get the output lined up exactly how you wish. And while not documented (at least, I did not see it) you can use the TITLES and HEADING portion of "LISTS" to output printer control characters to your printer. "LABELS" allows you to format your output in a different manner; you can for instance specify up to 32 labels accross, with a left margin of 0-999, 0-999 columns between labels, and 0-999 lines between labels. Using a trailing ";" concatenates adjacent field data and prints out a comma and a space. A trailing "&" concatenates and puts a space. Of special significance are the DIF routines and the SynFile+ to AtariWriter abilities. DIF (Data Interchange Format) allows for the flow of data between SynFile+ and other programs. For instance, You can send data files from SynFile+ to SynCalc or Visicalc. Or can load data into your SynFile+ from either. In addition, you can input data into SynFile+ from ANY source if that data is put into DIF format. For example, Mark Spires and I have written a Filemanager 800+ to DIF converter that allows you to transfer files from Filemanager 800+ to SynFile+ without having to retype the data. This program is Public Domain and available upon request. If demand is there, it will be made available to our Library. Last, but far from least, is the ability of SynFile+ to do database merges with AtariWriter. The story we have heard goes that

Continued on page 16

when AtariWriter was presented to Atari management for approval they decided that it was too complex and wanted some features deleted. Rather than delete the features, reference to them was simply left out of the documentation. SynFile+ uses one of these otherwise undocumented features to allow a database merge resulting in, among other things, the possibility of mass mailings. It seems that the only limit may be your imagination.

Is SynFile+ the perfect program??!? Well, no, unfortuneately not. There are some features not available that I miss from Filemanager 800+, such as repeating fields. And, when in the "RETRIEVE" mode you must push the START button for the next record: there is no "automatic" option. It also can be a pain that, after creating and using a "LISTS" or "LABELS" format, if you go back to "RETRIEVE", perhaps to change some data, you must redo you output format. Unlike Filemanager 800+, that format is not kept in memory once you leave that section. If you are outputting some data and notice, for instance, that you found a wrong data item or perhaps forgot a field you wanted to output, or need to change ANYTHING, you must enter all the "LISTS" or "LABELS" format information again. (It can be a REAL PAIN).

So it's not the perfect program. But, in my humble opinion, it is by far the most advanced program for the Atari, bar none. You can feel free to disagree with that opinion, but until I see something better... In any case, it is a superior program and I recommend it highly to anyone who wants a truely good data management program. If they were all like this, NO ONE would consider the Atari "just a game machine".

Not any more.

ELIMINATING ERROR 164
adapted by Ray Maynard

Editor's note: The following item was adapted from the December 1983 issue of Houston Atari Computer Enthusiasts newsletter.

Have any of you had probleas with ERROR 164? This error says that a "file number mismatch" has occurred. When that happens, you are unable to use that file any more. Gary Sewell and Wes Newell of ACUGD have come up with a fix. In ATARI DOS there is a routine that check for file number mismatches when loading a program. That routine can be easily NO-OPed as follows. With the BASIC cartridge inserted, boot up the system. Then type

POKE 4148,234:POKE 4149,234

The "234" is a NO-OP instruction. Then go to DOS and write out the new DOS. You can use this method with double-density also. Their final comment is "This patch should have no side effects with any good disks, so use it on all your disks."

BACK ISSUES AVAILABLE

If you are a new member and are missing issues of Keeping PACE from before you joined, you can buy copies of back issues at $1.25 each. So, if you have missed some of the series of articles on Assembly Language programming, this is your chance to get a full set. Contact the editor, Steve James, for details.

Shoe — Jeff MacNelly

SynTrend

BY SYNAPSE, DISK, $69.95

Reviewed by Allen Hart

SynTrend was developed exclusively for Atari by SYNAPSE. It comes on two disks and requires 48K of memory and BASIC. *SynTrend* actually consists of two packages, *SynGraph*, a flexible graphics package authored by Brian Lee, and *SynStat*, a statistics package authored by Randy Lert and Ron Conley.

SynStat

Introduction. *SynStat* is a menu driven statistics package that allows you to perform descriptive analysis or regression analysis on data that you may enter via the *SynStat* editor, or that you may have created with *SynFile*, *SynCalc*, *SynGraph*, or any program that allows you to store data in Data Interchange Format (DIF) files.

If you are a statistician, that introduction may mean something to you, but if you are not, an attempt is made later in this review to briefly explain what descriptive analysis and regression analysis can do for you.

Data Manipulation. In order to do any statistical analysis, you must first have data to analyze. *SynStat* provides two ways to enter the data. The first way is with an editor that operates like a spreadsheet (e.g. *VisiCalc*, *SynCalc*). The spreadsheet may contain up to 12 columns of numeric data and a total of 1000 cells. The *SynStat* editor displays three columns at a time with 15 rows in each column, and allows you to change the columns and rows that are being displayed, by moving around through the spreadsheet with the arrow keys. You enter data into the spreadsheet by simply moving to the desired cell with the arrow keys, and typing in the number you want.

The second way to enter data for statistical analysis is by loading data that are stored in DIF files. DIF files can be created by *SynCalc*, *SynFile*, *SynGraph*, and software available from other manufacturers. If you have data in DIF files, *SynStat* provides a command that allows you to load that data directly into a *SynStat* spreadsheet.

Column Calculations. Besides the ability to simply enter or load data, *SynStat* allows you to perform calculations on the columns in a spreadsheet. You can add, subtract, multiply, or divide two columns, and store the result in a third column. You can also add a constant to a column, multiply a column by a constant, or take the natural logarithm of all the numbers in a column.

Descriptive Analysis. Once you have a spreadsheet filled with data, you are ready to perform a descriptive analysis. *SynStat* performs a descriptive analysis on a single variable (column of data) in a spreadsheet. After you select the desired variable, the results of the analysis are provided in a display which contains:

- The Number of Observations
- The Average of all the numbers in the column
- The Standard Deviation (The average amount that all numbers deviated from the average)
- The Variance (The Standard Deviation Squared.)
- The Standard Error
- The Maximum
- The Minimum
- The Range (Maximum - Minimum)

Once the results of the descriptive analysis are displayed, you have the option of printing them by simply pressing the OPTION key.

Regression Analysis. Before I describe the *SynStat* Regression Analysis capabilities, I must make an admission to all you non-statisticians. I am not going to try to explain in much detail what regression analysis is, because it has been over ten years since I studied statistics. I will briefly describe what capabilities *SynStat* has, and when possible I will explain what the analysis can be used for. If you want to know more, I recommend (as does the *SynStat* documentation) that you refer to a statistics textbook.

To quote the *SynStat* users manual, "Regression is the most common statistical technique utilized to confirm or deny a hypothesis concerning the relationship that exists between two or more variables". The tutorial on regression in the users manual provides an example that utilizes data on a salesmans telephone expenses, automobile expenses, and total sales in each month. Using regression analysis, the tutorial shows that it is possible to decide whether the amount of automobile expenses (travel to see customers), or phone expenses (calls to customers) are having a major impact on the amount of sales each month.

Data for regression analysis by *SynStat* is entered in the same way as data for descriptive analysis, that is via the *SynStat* editor. In order to perform the regression analysis, you simply select the variables that you want to analyze. *SynStat* presents a menu with the variables you currently have stored in your spreadsheet. First you select a single dependent variable (in the example: sales for the month). Then you may select from one to 11 independent variables. You select the variables by using the arrow keys to move the cursor to the desired variable on the menu, thus highlighting the variable name, and pressing the RETURN key. When you have selected all the desired variables, pressing the START key causes *SynStat* to run the regression analysis.

After *SynStat* has run the regresson analysis, it allows you to view the results on a set of five different displays:

1) Regression Coefficients
2) Analysis of Variance
3) Partial Correlation Coefficients Squared
   Dependent Variable with Independent Variables
4) Correlation Matrix of Independent Variables
5) Residual Analysis

The **Regression Analysis** calculates the equation of a curve that fits the data points you have entered for analysis. The **Coefficient** display shows the coefficients of the calculated equation as well as a measure of how well the equation fits the data. The **Analysis of Variance** provides measures of how well the regression explains the process under study (e.g., the effect of the salesmans travel on his sales). The **Partial Correlation Coefficients** are meaningful only when a multiple regression analysis is done, and show how each independent variable correlates with the dependent variable when the other independent variables are held constant. The **Residual Analysis** shows a side by side comparison of the actual dependent variable values, and the predicted values based upon the independent variables. The **Correlation Matrix** of independent variables shows how the independent variables correlate to each other, which is another useful analysis tool when doing multiple regressions.

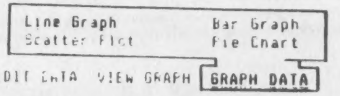
The *SynStat* software is very easy to use, and the tutorial provides a nice introduction to the software. For the more serious statistician, there is one drawback that I noticed. The documentation does not provide any mathematical basis for the results that are displayed. I believe the equations for regression analysis are reasonably standard, but a section of the manual for advanced users that showed the equations that are used in the *SynStat* analysis would have been a nice addition.

SynGraph

Introduction. *SynGraph* is a very useful and versatile graphics package that allows you to produce line graphs, bar graphs, pie charts, and scatter plots. *SynGraph* allows you to define many different characteristics of each of the graphs. It produces the graphs first on the computer screen in GRAPHICS 8. It then allows you to print the graph (to and Epson or a Prowriter), save the graph in a file on disk, or revise some of the characteristics, and have a new graph displayed.

Editor. The editor in *SynGraph* is much simpler than the one in *SynStat*. The *SynGraph* editor simply allows you to create (or edit) a DIF file. It displays a menu on the screen that allows you to enter numeric data for a single variable. You may enter up to 15 values on each screen, and may then advance to another screen via the START key. The editor allows you to store up to 100 values in a single DIF file.

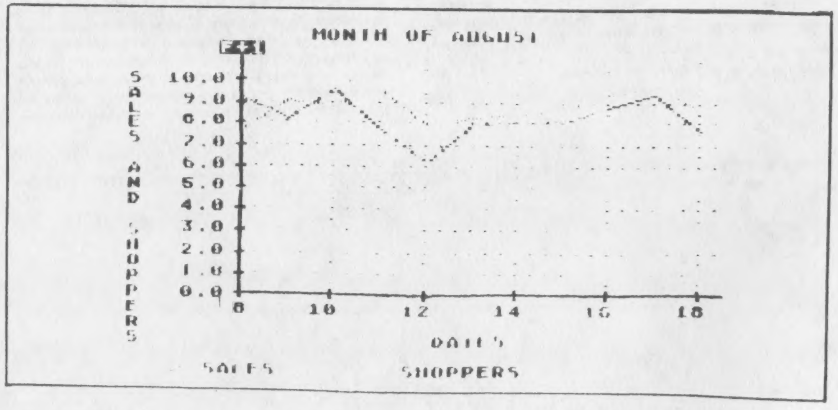
Selection of Graph Type. In order to select the type of graph you want to produce, you first select the GRAPH DATA option on the main menu. *SynGraph* then displays a selection menu on the bottom of the screen as shown below:



The highlighted entries in the menu are actually in reverse video on the screen. Using the arrow keys you may move around on the menu. When the desired type of graph is highlighted, you simply press RETURN. *SynGraph* then loads the correct program for that type of graph.

After you select a type of graph, you are shown a sequence of parameter entry screens that allow you to define the characteristics of the graph you want to produce. The types of characteristics you may specify include the title, the number of variables (factors) to be graphed, the labels to be used on the graph, the files to be used as a data source, the type of grid to be used on the graph, scaling information, and a file name to save the display in.

Line Graphs. A line graph is a graph of two variables on an x-y chart, with lines connecting each of the points. *SynGraph* allows you to graph up to three pairs of variables on one graph. Each pair of variables may consist



of up to 100 values each. A set of example graphs, produced by the *SynGraph* software are shown with this review. The first example is a sample line graph showing the number of shoppers and the number of sales in a hypothetical store, from the 8th through the 18th of August.

SynGraph allows you to define a line graph with a sequence of three parameter entry displays. A sample of the first display is shown below:

```
                LINE GRAPH
DISK DRIVE NUMBER: _

   TITLE OF GRAPH: ----------
                   ----------

NUMBER OF FACTORS: _

 NAMES OF FACTORS: --------
                   --------
                   --------

      Y-AXIS LABEL: ----------
      X-AXIS LABEL: ----------

    GRID (H,V,B,N): _

     FILE FOR SAVING: ---------
 OPTION=MAIN MENU  START=CONTINUE
```

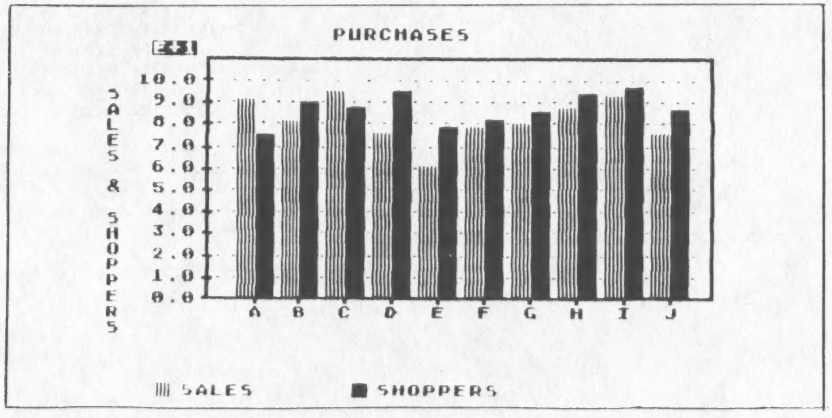
After you fill in the fields in the first display and press START, *SynGraph* displays a menu which lists all the DIF files on your data disk. You are then prompted to enter two file names for each factor you have named in the first parameter entry display. You must enter a file name for the x variable, and a file name for the y variable.

SynGraph next reads the data files you have specified, and displays a new parameter entry screen which allows you to scale your graph. The format of the display is shown below:

```
                LINE GRAPH
                SCALING
 AXIS  CURRENT MIN  CURRENT MAX
 ----  -----------  -----------
  Y       xx.xx        xx.xx
  X       xx.xx        xx.xx
-------------------------------

  REVISED MIN Y: ----------
  REVISED MAX Y: ----------

  REVISED MIN X: ----------
  REVISED MAX X: ----------

 Y DIVISIONS (1-15): __
  X DIVISIONS (1-6): _

  INTEGER (X,Y,B,N): _

OPT=ABORT SLCT=AUTOSCALE START=CONT
```

You have the option of revising the minimum and maximum scale values that are to be used on the graph, specifying the number of divisions you want annotated on each axis, and indicating if you want the annotations on the x and/or y axes to be integers rather that real numbers. If you want, you can have *SynGraph* automatically scale the graph for you by simply hitting the SELECT key.

Scatter Plots. Scatter plots are almost identical to line plots, with the single exception that the data points



that are plotted on the x-y graph are not connected by lines. Scatter plots are created in exactly the same way as line graphs. The menus displayed by *SynGraph* for scatter plots are identical to those for line graphs, with the single exception of the title line on the menu.

Bar Graphs. Bar graphs are very useful graphs that allow the comparison of the magnitudes of two or three variables, or the illustration of a trend in a single variable. The example graph contains the same data that was used in the line graph example. The example demonstrates one of the two types of bar charts supported by the *SynGraph* package, stacked bar graphs, and clustered bar graphs.

SynGraph allows you to produce bar graphs containing up to 32 single-factor bars, 24 two-factor bars, or 16 three-factor bars. Clustered bar graphs can display both positive and negative values, but stacked bar graphs can display only positive values.

Bar graphs are defined to *SynGraph* in almost the same way as line graphs. The only difference in the first menu is that bar graphs allow only horizontal grids, so the menu does not allow for vertical grids or grids in both directions. The data selection menu and scaling menu are similar too, except that you only scale data in the y direction for bar graphs. One big difference for bar graphs, is the addition of a menu to define labels for each of the bars (or clusters of bars). One useful application of the labelling is to specify the months of the year under each bar.

Pie Charts. Pie charts are a useful graphing technique to show the percentage that each major component makes up in a total package. They are used very frequently to show how the money in a budget is divided among the different components of the budget. A pie chart is shown in the set of example graphs.

SynGraph will handle pie charts with up to 12 slices. It calculates the percentages to be displayed beside each slice, and allows you to enter a label of up to seven characters to be displayed by each slice.

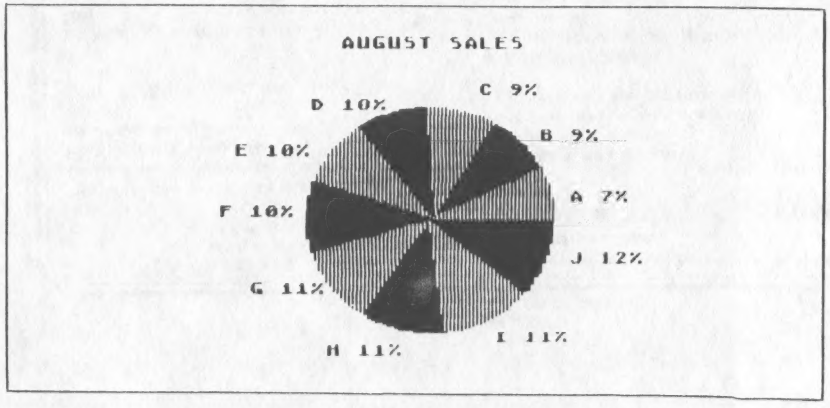
There are only three menus for the generation of pie charts. The first menu allows you to specify a disk drive, a graph title, and a file to save the display in. The second menu allows you to select the data file to be used as the data source. The last menu allows you to specify the labels to be used on each slice of the pie.

Printing. In addition to displaying the graphs on the screen, *SynGraph* will print the graphs on an EPSON, or a PROWRITER. All that is required to print the displays is to hit the OPTION key, use the arrow keys to select the printer type, and hit the RETURN key. If you are like me, and you don't have an EPSON, or a PROWRITER, you can still get printed copies of the displays, as long as you have a printer with graphics capability, and a BASIC program that will dump a GRAPHICS 8 screen to your printer.

Saving. In addition to the ability to print your graphs, *SynGraph* allows you to save a copy of the graph in a disk file, and call the graph back up for later viewing.

The user manual contains the listing of a program that will generate a slide show of graphs by reading a set of saved graphs from the disk and displaying them on the screen. If you have a BASIC routine to dump a GRAPHICS 8 screen to your printer, you can modify the slide show program to first load the displays from the disk, and then print them on your printer. I really am sure you can, because that is how the example graphs in this article were produced.

Conclusions. The *SynTrend* software package is a professionally produced product. It is easy to use, and flexible. Both the statistics package and the graphics package should prove to be valuable tools for anyone who is interested in data trend analysis.



Atari Basic's TRAP Command

Paul Maternowski HANKATARI NOV 84

The TRAP command allows you to re-direct program flow in a BASIC program. It allows you to direct what a program will do when it generates a error condition. Take a look at the program below for a sample of it's use:

```
 5 DIM LINE$(255)
10 TRAP 1000
20 OPEN #1,4,0,"D:FILENAME"
30 TRAP 2000
40 INPUT #1,LINE$
50 ? LINE$
60 GOTO 40
1000 CLOSE #1:?"Could not find file"
1020 END
2000 CLOSE #1:?"File read successfully"
```

This program will read the file "D:FILENAME" and print it to the screen. The TRAP instruction in line 10 tells BASIC to goto line 1000 on the next error. If line 20 generates a error condition from not having the disk with that filename in the drive, a goto line 1000 will be executed telling you that file is not on the disk. Line 30 sets the trap to line 2000. Lines 40,50 & 60 will be executed until an end-of-file error occurs, at which time a goto line 2000 will be executed. The file read successfully message will be printed to the screen.

The TRAP can be turned off by using a TRAP number greater than 32767. When you type in an immediate mode line such as:

RUN<RETURN> or ?2+2<RETURN>

you are actually typing in with a line number of 32768. The TRAP instruction takes the line number after it and stores it in two bytes at locations 188 and 189. When a error occurs in BASIC, it examines location 189 to check if the number is less than 128 which corresponds to a line number less than 32768. If the number in this location is less than 128, BASIC instead of printing out ERROR XXX etc., will instead goto the line number stored in locations 188 and 189. Once a TRAP is executed, it is cleared and the TRAP instruction must be set again if it is to be reused.

I/O Connector Source

A good source of ATARI 13 pin I/O connectors is Centurian Enterprises at $3.00 apiece. They stock both types, internal and the cable end types, along with a complete source of parts, kits and software at good discounts. If interested write for a catalog: P.O.Box 3233, San Luis Obispo, CA 93403-3233

VISICALC
QUICKY GUIDE
by M. Aldrich

Have you ever been working along with your VISICALC and just plain can't remember how to do something. Well here's a little quicky guide I've put together. Hope you get some use out of it.

1. BOOT UP PROCEDURE: Boot from drive #1. Wait 2 minutes.

2. DISK INITIALIZATION: Insert disk. Press / S I and Return.

3. CLEAR SPREADSHEET: / C Y

4. GET SERIAL# SCREEN: / V

5. CURSOR MOVEMENTS: Press Control and Inverse arrow key.

6. DIRECT CURSOR MOVEMENT: Press > (col10) Return (example >B3).

7. BACKSPACE-BACKOUT: Press Backspace # times needed.

8. QUICK CALCULATION: Example press 25 * 25 !

9. ENTER & MOVE: Type in Label or Value and then CTRL Arrow.

10. MATH OPERATORS: Mult - *, Div - /, Add - +, Sub - -

11. BORDERS-LIMITS: Right - BK254, Bottom - A254

12. LABELS: Type letters or precede label with a " for other illegal label symbols. Can be as long as you like.

13. VALUES-FORMULA: Type Number or Formula or + and cell name. A + is needed only if a cellname will be used for the first item in the formula.

14. Cells: You can't See! Press +, move cursor to cell you want, press then next operator (example -), move cursor to next cell, press next operator, etc...

15. SAVING FILES: Press / S S Type filename (VC is appended.

16. BLANK AN ENTRY: Press / B Return.

17. LOADING FILES: Press / S L Return.

18. SCROLLING THRU FILES: Press / S L Right arrow Right Arrow etc...

19. FORMATTING SCREEN DISPLAY: Press / G F I (Global Format Integer).

20. FORMATTING SCREEN DISPLAY: Press / G F $ (Global Format is Dollars & Cents).

21. ADJUSTING COLUMN WIDTHS: Press / G C 7 (For column width of 7).

   a. 3 to 37 width available.
   b. Can only do globally.
   c.)) appears if no space between cells.
   d. Labels don't have this extra space.

22. SPLITTING THE SCREEN:
   a. Type >E1 (Splits window on the left of E!)
   b. Type / W
   c. Type V
   d. Type ; (To Jump back & forth between screens).
   e. Type /GC4 (Only affects window where cursor is at.)

23. Going back to 1 window: Type / W 1

24. SPLITTING THE SCREEN: Horizontally:
   a. Go back to 1 window process.
   b. Type > B11
   c. Type / W H
   d. Type ; (to jump back & forth)

25. FIXING TITLES IN PLACE:
   a. Move cursor to left edge (on titles >A1)
   b. Type / T V CTRL Right arrow (8 times) or >B1
      1. H to fix horizontal titles
      2. V to fix vertical titles
      3. B to fix both
      4. N to fix neither
   c.)(cell thats a title) to get cursor on that cell (single cursor moves won't do it.

26. RETURNING THINGS TO NORMAL:
   a. Type / W 1 / T N / G F G / G C 9 return.

27. REPLICATING:
   a. Drawing lines across the sheet
      1. Type >A3 / -
      2. Type - return
      3. Type / R return
      4. Type B3.M3 return

   b. Replicating numbers & labels:
      1. Type INCOME in A2
      2. Type 1770 in A2
      3. Type / R return
      4. Type C2.M2 return

   c. Replicating another (just 1 line)
      1. Type in a label at A1
      2. Type a number at B1
      3. Type a formula at C1
      4. Type / R return
      5. Type B1.M1 return
      6. Type R